

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra telekomunikační techniky

**Možnosti sazby elektronických schémat  
v systému L<sup>A</sup>T<sub>E</sub>X a PGF/TikZ**

**DTP of Electronic Schematics using  
PGF/TikZ**

# Zadání bakalářské práce

Student:

**Martin Kuběna**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2601R013 Telekomunikační technika

Téma:

Možnosti sazby elektronických schémat v systému LaTeX a PGF/TikZ  
DTP of Electronic Schematics using PGF/TikZ

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je popsat možnosti sazby elektronických schémat s využitím systému LaTeX a PGF/TikZ pro typografický systém LaTeX.

1. Proveďte rešerši balíčků pro sazbu elektronických schémat v systému LaTeX a PGF/TikZ.
2. Proveďte rešerši WYSIWYG editorů pro sazbu elektronických schémat v systému LaTeX a PGF/TikZ na platformách Windows a Linux.
3. Pro zvolené balíčky (po konzultaci se vedoucím práce) vytvořte podrobný návod pro sazbu elektronických schémat.
4. Po konzultaci s vedoucím práce popište detailně sazbu některých vybraných schémat z oblasti telekomunikačních systémů.
5. Proveďte subjektivní hodnocení práce s jednotlivými nástroji.

Seznam doporučené odborné literatury:

[1] GAJDOŠÍK, Libor. *Návod na použití makra eltex 2.0* [online]. [cit. 2016-10-17]. Dostupné z: <https://www.ctan.org/tex-archive/macros/latex/contrib/eltex>

[2] PAVLOREK, Tomáš. *Použití balíčku TikZ pro sazbu LaTeXových knih s obrázky*. Ostrava, 2013. Bakalářská práce. Vysoká škola báňská - Technická univerzita Ostrava. Fakulta elektrotechniky a informatiky. Vedoucí práce Jan Skapa. URL: <http://dspace.vsb.cz/handle/10084/98929>

[3] TANTAU, Till. *The TikZ and PGF Packages Manual* [online]. [cit. 2016-10-17]. Dostupné z: <http://ctan.mackichan.com/graphics/pgf/base/doc/pgfmanual.pdf>

[4] OUTRATA, Jan. *Diplomová propedeutika* [online]. [cit. 2016-10-17]. Dostupné z: <http://outrata.inf.upol.cz/courses/dp/>

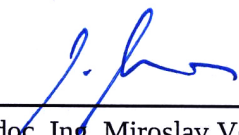

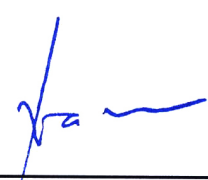
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Skapa, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 30.04.2018

---

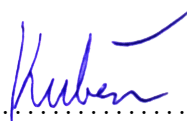
  

doc. Ing. Miroslav Vozňák, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

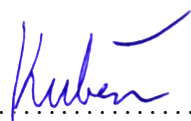
Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 25. června 2018

.....  


Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 25. června 2018

.....

Rád bych na tomto místě poděkoval především vedoucímu své práce Ing. Janu Skapovi, Ph.D. za jeho cenné rady a čas strávený při konzultaci této práce.

## **Abstrakt**

Cílem práce je popsat možnost sazby elektronických schémat v typografickém systému L<sup>A</sup>T<sub>E</sub>X a PGF/TikZ. Nejprve řeší dostupných balíčků a maker pro pokročilé sázení s nutnou znalostí syntaxe, která bude u určitých balíčků podrobně vysvětlena a předvedena na vybraném schéma z oblasti telekomunikačních systémů. Následně přehledem a popisem dostupných WYSIWYG editorů, které sazbu maximálně zjednoduší pro začínající uživatele. Na závěr bude provedeno zhodnocení jednotlivých nástrojů a práce s nimi.

**Klíčová slova:** T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, PGF/TikZ, elektrické schémata

## **Abstract**

The aim of the thesis is to describe the possibility of electronic scheme rates in the typographic system L<sup>A</sup>T<sub>E</sub>X and PGF/TikZ. Firstly, search for available packages and advanced betting macros with the necessary syntax knowledge, which will be explained and demonstrated in detail on certain packages in the selected telecommunication scheme. Subsequently, a review and description of available WYSIWYG editors, which will make the rate as simple as possible for novice users. Finally, the tools will be evaluated and worked with them.

**Key Words:** T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, PGF/TikZ, electrical diagrams

# Obsah

<b>Seznam použitých zkratek a symbolů</b>	<b>10</b>
<b>Seznam obrázků</b>	<b>11</b>
<b>Seznam výpisů zdrojového kódu</b>	<b>12</b>
<b>1 Instalace L<sup>A</sup>T<sub>E</sub>Xu</b>	<b>14</b>
1.1 Windows . . . . .	14
1.2 Linux . . . . .	15
<b>2 MikTeX</b>	<b>16</b>
2.1 Instalace Windows . . . . .	16
2.2 Instalace Linux . . . . .	16
<b>3 Knihovny pro sazbu</b>	<b>18</b>
3.1 CircuiTikZ . . . . .	18
3.2 Circuits . . . . .	18
3.3 elTeX . . . . .	19
3.4 Circ . . . . .	24
3.5 pst-circ . . . . .	26
3.6 Latex Circuit Diagram . . . . .	29
3.7 CircDia . . . . .	31
<b>4 WYSIWYG editory</b>	<b>35</b>
4.1 Dia . . . . .	35
4.2 TikzEdt . . . . .	36
4.3 Inkscape . . . . .	40
<b>5 Návod pro práci s balíčky</b>	<b>43</b>
5.1 Circuitikz . . . . .	43
5.2 Latex Circuit Diagram . . . . .	51
<b>6 Závěr</b>	<b>57</b>
6.1 Circuitikz . . . . .	57
6.2 Circuits . . . . .	57
6.3 elTeX . . . . .	57
6.4 Circ . . . . .	58



6.5	pst-circ . . . . .	58
6.6	Latex Circuit Diagram . . . . .	58
6.7	CircDia . . . . .	58
<b>7</b>	<b>Reference</b>	<b>60</b>
	<b>Literatura</b>	<b>60</b>
<b>8</b>	<b>Přílohy</b>	<b>62</b>
	<b>Přílohy</b>	<b>62</b>

## Seznam použitých zkratek a symbolů

PGF	– Portable Graphics format
TikZ	– TikZ ist kein Zeichenprogramm
PDF	– Portable Document Format
EPS	– Encapsulated PostScript

## Seznam obrázků

1.1	Rozložení editoru Texmaker . . . . .	14
2.1	MiKTeX Package Manager s filtrací makra Circ . . . . .	17
3.1	Rozdíl mezi absolutními a relativními pozicemi popsany balíčkem Circuitikz . . .	19
3.2	Náhradní schéma vedení v CircuiTikz . . . . .	20
3.3	Schéma normálového generátoru v CircuiTikz . . . . .	21
3.4	Náhradní schéma vedení v Circuits . . . . .	22
3.5	Schéma normálového generátoru v Circuits . . . . .	23
3.6	Náhradní schéma vedení v elTeX . . . . .	25
3.7	Náhradní schéma vedení v makru Circ . . . . .	27
3.8	Schéma normálového generátoru v makru Circ . . . . .	28
3.9	Náhradní schéma vedení v makru pst-circ . . . . .	30
3.10	Náhradní schéma vedení vytvořeného pomocí Latex Circuit Diagram . . . . .	32
3.11	Možnosti různého uspořádání textu součástek u makra CircDia . . . . .	33
3.12	Náhradní schéma vedení vytvořeného pomocí balíčku CircDia . . . . .	34
4.1	Rozhraní editoru Dia . . . . .	36
4.2	Náhradní schéma vedení kreslené v Dia . . . . .	36
4.3	Schéma tranzistorového zesilovače kreslené v Dia . . . . .	37
4.4	Schéma normálového generátoru kreslené v Dia . . . . .	37
4.5	Práce v editoru TikzEdt . . . . .	38
4.6	Náhradní schéma vedení vytvořené v TikzEdt s knihovnou Circuits . . . . .	39
4.7	Schéma normálového generátoru vytvořené v TikzEdt s knihovnou Circuits . . .	39
4.8	Schéma tranzistorového zesilovače vytvořené v TikzEdt s knihovnou Circuits . .	39
4.9	Okno rozšíření CircuitSymbols pro nastavení a vložení RLC prvků . . . . .	41
4.10	Náhradní schéma vedení kreslené v Inkscape . . . . .	41
4.11	Schéma tranzistorového zesilovače kreslené v Inkscape . . . . .	42
4.12	Schéma normálového generátoru kreslené v Inkscape . . . . .	42
5.1	Použití parametru pro vyznačení uzlů a kontaktů . . . . .	46
5.2	Vykreslení popsaného NPN tranzistoru . . . . .	47
5.3	Schéma invertujícího zesilovače . . . . .	47
5.4	Schéma nf zesilovače v Circuitikz . . . . .	50
5.5	Úprava vertikálního kondenzátoru pro druhý popis . . . . .	54
5.6	Schéma nf zesilovače v Latex Circuit Diagram . . . . .	55

## Seznam výpisů zdrojového kódu

3.1	Náhradní schéma vedení v CircuiTikz . . . . .	20
3.2	Schéma normálového generátoru v CircuiTikz . . . . .	21
3.3	Náhradní schéma vedení v Circuits . . . . .	22
3.4	Schéma normálového generátoru v Circuits . . . . .	23
3.5	Náhradní schéma vedení v elTeX . . . . .	25
3.6	Náhradní schéma vedení v makru Circ . . . . .	27
3.7	Schéma normálového generátoru v makru Circ . . . . .	28
3.8	Prostředí pspicture s velikostí mřížky 11,5 . . . . .	29
3.9	Náhradní schéma vedení v makru pst-circ . . . . .	30
3.10	Náhradní schéma vedení v makru Latex Circuit Diagram . . . . .	32
3.11	Náhradní schéma vedení v makru CircDia . . . . .	34
5.1	Příklady zápisu normálních prvků . . . . .	44
5.2	Příklady zápisu bipólových prvků . . . . .	44
5.3	Příklady použití názvů a hodnot prvků . . . . .	45
5.4	Použití absolutních souřadnic . . . . .	46
5.5	Použití relativních souřadnic . . . . .	46
5.6	Schéma normálového generátoru v CircuiTikz . . . . .	47
5.7	Příklady syntaxe a orientace prvků . . . . .	53

### Úvod

L<sup>A</sup>T<sub>E</sub>X (čti Latech) je typografickým, volně šiřitelným systémem, který je určen k sazbě vědeckých a matematických dokumentů vysoké typografické kvality. Systém je rovněž vhodný pro tvorbu všech možných druhů jiných dokumentů, od jednoduchých dopisů po složité knihy. Systém L<sup>A</sup>T<sub>E</sub>X je postaven na typografickém formátovacím programu T<sub>E</sub>X Donalda E. Knutha.

Pomocí tohoto systému je možné vytvořit soubor, obsahující text dokumentu a do něj vkládat speciální příkazy, které nám zaručí, jak má tento text být formátován. Ačkoliv je T<sub>E</sub>X výkonným systémem, je pro běžné použití příliš složitý a proto byly nad ním vytvořeny nadstavby, které umožňují snadnější a přirozenější zápis sázeného textu. Nejznámější nadstavbou je právě L<sup>A</sup>T<sub>E</sub>X vytvořený americkým počítačovým vědcem Leslie Lamportem, který poměrně složitý T<sub>E</sub>X zjednodušil pomocí množiny maker a šablon a vytvořil poměrně snadný systém, jenž si časem získal popularitu a s ní i další rozšíření a funkce.[1]

Právě ona rozšíření nám umožňují v dnešní době sázet profesionálně vykreslená elektronická schémata. Možnosti jejich sazby budou obsahem právě této bakalářské práce.

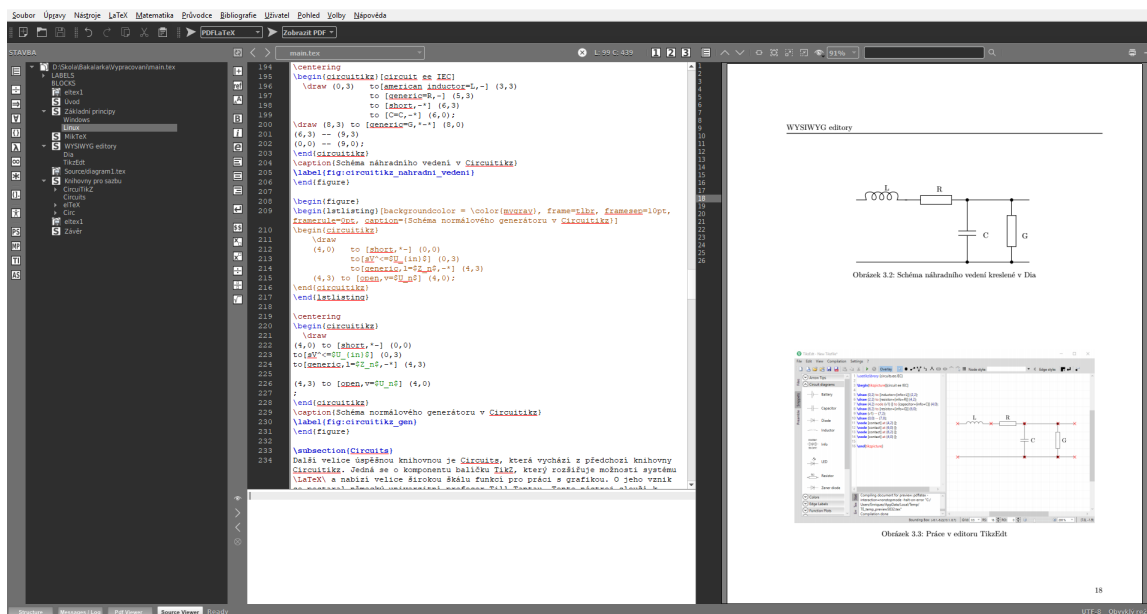
## 1 Instalace L<sup>A</sup>T<sub>E</sub>Xu

Abychom mohli začít sázet elektronická schémata v systému L<sup>A</sup>T<sub>E</sub>X, musíme si nejprve nainstalovat samotný L<sup>A</sup>T<sub>E</sub>Xa editor pomocí kterého budeme sázet, editovat a kompilovat zdrojový kód. Instalace je možná jak pro platformu Windows, tak i pro Linux.

## 1.1 Windows

Pro nejjednodušší instalaci a následnou správu nainstalovaných komponent je pro obě platformy software MikTeX2. Pokud však požadujeme pouze základní instalaci, stáhneme si ze stránek [tug.org](http://tug.org)<sup>1</sup> nástroj T<sub>E</sub>XLive a spustíme instalaci. Při poměrně dlouhé instalaci se mezitím můžeme rozhodnout, který editor bude pro nás nejlepší.

Mezi špičku patří editory Texmaker<sup>2</sup> a TeXnicCenter<sup>3</sup>, které obsahují velké množství funkcí, přívětivé prostředí a jsou zdarma. Texmaker nabízí také českou lokalizaci a možnost zobrazení výsledného dokumentu přímo v okně programu. TeXnicCenter je svým vzhledem a rozložením funkčních ikon vhodný pro uživatele MS Office.



Obrázek 1.1: Rozložení editoru Texmaker

<sup>1</sup>[www.tug.org/texlive/acquire-netinstall.html](http://www.tug.org/texlive/acquire-netinstall.html)

<sup>2</sup><http://www.xmlmath.net/texmaker/download.html>

<sup>3</sup><http://www.texniccenter.org/download/>

## 1.2 Linux

Pokud je nám bližší Linux, musíme stejně jako u Windows nejprve nainstalovat  $\text{\TeX}$ Live a podporu češtiny a slovenštiny pomocí příkazů:

```
sudo apt-get install texlive
sudo apt-get install texlive-lang-czechslovak
```

Následuje instalace editoru, kterým je stejně jako u Windows nejvhodnější Texmaker nebo alternativa o něco jednodušší Gummi<sup>4</sup>, který podporuje realtime zobrazování změn ve výsledném dokumentu:

```
sudo apt-get install texmaker
sudo add-apt-repository ppa:gummi/gummi
sudo apt-get update
sudo apt-get install gummi
```

Pro tvorbu bakalářské práce jsem použil textový editor a prohlížeč výstupních PDF souborů Texmaker a sázecí systém MiKTeX jako správce nainstalovaných balíčků a maker.

---

<sup>4</sup><https://github.com/alexandervdm/gummi>

## 2 MikTeX

MiKTeX<sup>1</sup> je software obsahující kompilátor TeXu, LaTeXu a spřízněných variant. Pracuje převážně na platformě Microsoft Windows, ale lze jej použít i pro platformu Linux vyšších verzí, vyvíjí jej Christian Schenk. Jeho součástí je také textový editor TeXworks určen pro základní upravování zdrojového textu. Přes svoji velmi jednoduchou instalaci je také zajímavý z mnoha důvodů[2]:

- Přehledný **Package Manager**, kde můžeme velice snadno vyhledávat, editovat, ale také instalovat nové balíčky a makra.
- Automatické stahování chybějících balíčků při kompilaci dokumentu
- Jednoduchou aktualizaci balíčků a maker pomocí **Update Wizard**
- Rychlé náhledy při cyklu *editace - kompilace - shlédnutí* umožňuje **DVI Previewer (Yap)**

Pro sazbu elektronických schémat bude pro nás užitečná správa nainstalovaných balíčků, ale především zjednodušená instalace některých maker.

### 2.1 Instalace Windows

Instalátor pro systém Windows je ke stažení z webových stránek MiKTeXu<sup>2</sup> ve verzi 2.9. Na výběr máme základní instalaci, internetovou instalaci a portable instalaci na externí zařízení. Ve všech případech se jedná o instalaci všech potřebných komponent a základních balíčků. Po dokončení instalace je doporučeno spustit **MiKTeX Console** a zkontrolovat, zda nejsou dostupné aktualizace v záložce *Updates* Nové balíčky můžeme doinstalovat ručně v **Package Manageru** nebo je nechat stáhnout automaticky při kompilaci dokumentu, který je obsahuje.

### 2.2 Instalace Linux

Pro Linux nabízí MiKTeX instalaci pro verze Ubuntu 16.04, Ubuntu 18.04, Linux Mint 18 nebo vyšší a Debian 9. Před samotnou instalací se nejprve zaregistruje úložiště, které obsahuje MiKTeX instalační balíček. Pro Ubuntu 18.04 příkazem:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys D6BC243565B
2087BC3F897C9277A7293F59E4889
echo "deb http://miktex.org/download/ubuntu bionic universe"| sudo tee /etc/apt/
```

---

<sup>1</sup><https://miktex.org/>

<sup>2</sup><https://miktex.org/download>



## MiKTeX

---

```
sources.list.d/miktex.list
```

```
sudo apt-get update
```

Po registraci můžeme přejít k instalaci:

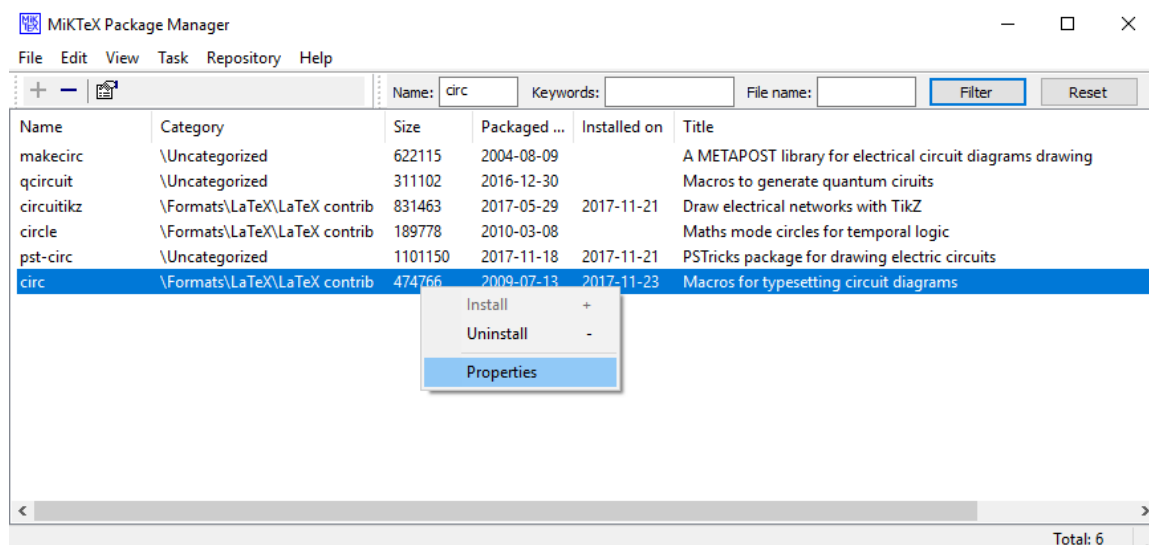
```
sudo apt-get install miktex
```

Instalaci dokončíme spuštěním **MiKTeX Console**, kde zvolíme zda se bude jednat o privátní instalaci nebo sdílenou pro všechny uživatele. V příkazové řádce pro privátní instalaci:

```
miktexsetup finish
```

A sdílenou instalaci:

```
sudo miktexsetup -shared=yes finish
```



Obrázek 2.1: MiKTeX Package Manager s filtrací makra Circ

### 3 Knihovny pro sazbu

Pro sazbu elektrických schémat v systému L<sup>A</sup>T<sub>E</sub>X je nezbytná některá z knihoven, kterou musíme nejprve (pokud již není obsažena v instalaci T<sub>E</sub>Xu) nainstalovat a poté nahrát do hlavičky našeho dokumentu. Každá knihovna má svoji specifickou syntaxi sazby popsanou ve svém manuálu, kterou musíme dodržovat. V manuálu pro danou knihovnu také najdeme základní popis, návod pro případnou instalaci, přehled všech komponent které knihovna nabízí a samozřejmě příklady sazby jednodušších schémat. Při sazbě je také doporučeno dbát na používání pouze jedné knihovny a to jak z estetického hlediska, tak z hlediska možných duplicit v externích souborech a tudíž i chyb při kompilaci souboru.

Knihovny můžeme rozdělit podle způsobu vkládání komponent na absolutní nebo relativní - při absolutním vkládání určuje pozici imaginární mřížka (můžeme ji samozřejmě pro lepší orientaci také vykreslit), do které dosazujeme komponenty podle X,Y souřadnic. Oproti tomu relativní vkládání určuje pozici prvku od pozice prvku předchozího, imaginární mřížka zde tedy není potřeba.<sup>3.1</sup>

#### 3.1 CircuiTikZ

Zřejmě nejznámější a nejobsáhlejší knihovnu pro sazbu elektronických schémat v systému L<sup>A</sup>T<sub>E</sub>X vytvořil Massimo Redaelli v roce 2007 na univerzitě v Miláně. Po jeho odchodu z akademického prostředí v roce 2010 se výzkum zpomalil až do roku 2015, kdy se k němu připojili Stefan Lindner a Stefan Erhardt, kteří společně udržují projekt dodnes.<sup>[3]</sup>

Použití knihovny je široce dostupné pod licencí GNU General Public License a The L<sup>A</sup>T<sub>E</sub>X Project Public License.<sup>[3]</sup>

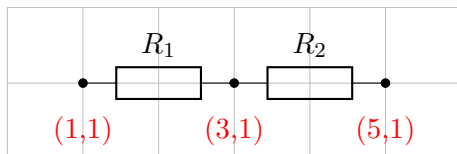
Knihovnu CircuiTikz do našeho dokumentu načteme pomocí příkazu:

```
\usepackage{circuitikz}
```

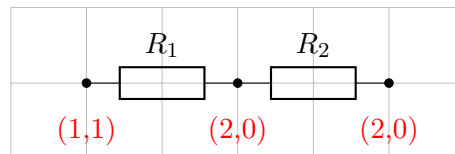
#### 3.2 Circuits

Další velice úspěšnou knihovnou je Circuits, která vychází z předchozí knihovny Circuitikz. Jedná se o komponentu balíčku TikZ, který rozšiřuje možnosti systému L<sup>A</sup>T<sub>E</sub>X a nabízí velice širokou škálu funkcí pro práci s grafikou. O jeho vznik se postaral německý univerzitní profesor Till Tantau. Tento nástroj slouží k vytvoření velice kvalitních grafických ilustrací na bázi vektorové grafiky a také profesionálně vytvořených elektrických schémat.<sup>[4]</sup>

Při definování je dobré použít přímo knihovnu `CircuiTikz` a ne pouze `Tikz`, jelikož hodně příkazů vychází právě z knihovny `CircuiTikz`. Dále je nutné upřesnit, že budeme používat z balíčku evropské IEC komponenty. Nahrání knihoven v hlavičce bude tedy vypadat:



```
\begin{circuitikz}
\draw[help lines, color=lightgray]
  (0,0) grid (6,2);
\draw(1,1) to [R=$R_1$,*-*] (3,1)
to [R=$R_2$,-*] (5,1)
(1,0.7) node[below, color=red] {(1,1)}
(3,0.7) node[below, color=red] {(3,1)}
(5,0.7) node[below, color=red] {(5,1)}
};
\end{circuitikz}
```



```
\begin{circuitikz}
\draw[help lines, color=lightgray]
  (0,0) grid (6,2);
\draw(1,1) to [R=$R_1$,*-*] ++(2,0)
to [R=$R_2$,-*] ++(2,0)
(1,0.7) node[below, color=red] {(1,1)}
(3,0.7) node[below, color=red] {(2,0)}
(5,0.7) node[below, color=red] {(2,0)}
};
\end{circuitikz}
```

Obrázek 3.1: Rozdíl mezi absolutními a relativními pozicemi popsany balíčkem Circuitikz

```
\usepackage{circuitikz}
\usetikzlibrary{circuits.ee.IEC}
```

Circuits má velice podobnou syntaxi jako Circuitikz a na první pohled se liší pouze zápisem parametrů u komponent a přehlednějším zápisem uzlů. Výhodou je také možnost volby absolutního nebo relativního formátu souřadnic.

### 3.3 elTeX

elTeX je makro vytvořené pro sázení jednodušších elektronických schémat Dr. Ing. Liborem Gajdošíkem z Katedry telekomunikační techniky VŠB-TUO.

Pro zprovoznění makra nám stačí si stáhnout ze stránek ftp.cstug.cz soubor `eltex.sty` a umístit jej do složky kde se nachází již existující `bezier.sty`. Konkrétně v adresáři `\tex\latex\base`. Jako manuál s příklady různých zapojení si můžeme ze stejné webové stránky stáhnout taky soubory `eltdoc1.tex` a `eltdoc2.tex` pro český a anglický překlad.

#### 3.3.1 Práce s makrem

Makro nahrajeme do hlavičky podle komponent, které budeme k sázení používat. Celé makro je rozdělené do sedmi bloků `eltex1` až `eltex7`, kdy obsah každého bloku je popsán v dokumentaci makra. Pro nahrání jednotlivých bloků použijeme příkaz `\input`:

```
\input{eltex1}
```

```

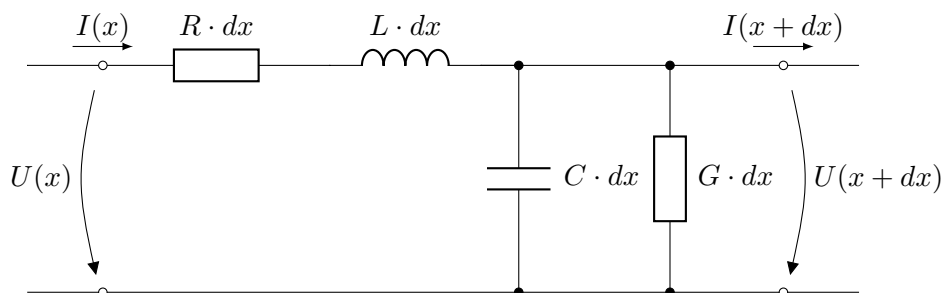
\begin{circuitikz}
  \draw (0,3) to [short] (1,3) % kaskada R-L-C
        to [R=$R \cdot dx$, -] (4,3)
        to [american inductor=$L \cdot dx$, -] (6,3)
        to [short, -*] (6.5,3)
        to [C=$C \cdot dx$, -*] (6.5,0);
  \draw (8.5,3) to [R=$G \cdot dx$, -*] (8.5,0) % G a propojovací vodice
        (6,3) -- (11,3)
        (0,0) -- (11,0);

  \draw[->, >=latex] (1,3) ++(up:0.25) ++(left:0.4) -- % Proud  $i(x)$ 
  node[left] {} ++(right:0.8);
  \node at (1,3.5) {$I(x)$};
  \draw[->, >=latex] (10,3) ++(up:0.25) ++(left:0.4) -- % Proud  $i(x+dx)$ 
  node[left] {} ++(right:0.8);
  \node at (10,3.5) {$I(x+dx)$};

  \draw (1,3) to [open, v=$U(x)$, o-o] (1,0); % Napětí  $U(x)$ 
  \draw (10,3) to [open, v^=$U(x+dx)$, o-o] (10,0); % Napětí  $U(x+dx)$ 
\end{circuitikz}

```

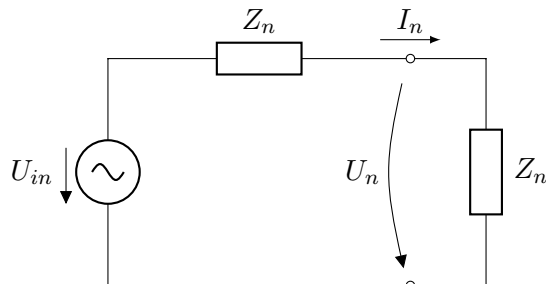
Výpis 3.1: Náhradní schéma vedení v CircuiTikz



Obrázek 3.2: Náhradní schéma vedení v CircuiTikz

```
\begin{circuitikz}
  \draw
    (4,0) to [short,o-] (0,0) % Kaskada  $U_{in}$ - $Z_n$ - $Z_n$ 
    to [sV^<= $U_{in}$ ] (0,3)
    to [ $R=Z_n$ ,o-] (4,3)
    to [short] (5,3)
    to [ $R=Z_n$ ] (5,0) -- (4,0)
    (4,3) to [open,v= $U_n$ ] (4,0); % Napeti  $U_n$ 
  \draw[->,>=latex] (4,3) ++(up:0.25) ++(left:0.4) -- % Proud  $I_n$ 
  node[left] {} ++(right:0.8);
  \node at (4,3.5) { $I_n$ };
\end{circuitikz}
```

Výpis 3.2: Schéma normálového generátoru v CircuiTikz



Obrázek 3.3: Schéma normálového generátoru v CircuiTikz

```

\begin{tikzpicture}[circuit ee IEC]
\node (con1) [contact] at (1,2) {}; % Definovani jednotlivych uzlu
\node (con2) [contact] at (1,0) {};
\node (con3) [contact] at (5,2) {};
\node (con4) [contact] at (5,0) {};
\node (con5) [contact] at (7,2) {};
\node (con6) [contact] at (7,0) {};
\node (con7) [contact] at (9,2) {};
\node (con8) [contact] at (9,0) {};

\draw (con1) to [resistor={info=$R \cdot dx$}] (3,2) % Kaskada R-L-C
to [inductor={info=$L \cdot dx$}] (con3)
to [capacitor={info=$C \cdot dx$}] (con4);
\draw (con5) to [resistor={info=$G \cdot dx$}] (con6); % G a propojovací vodice
\draw (0,0) -- (10,0)
(con3) -- (10,2)
(0,2) -- (con1);
\draw[fill=white] (con1) circle (1.7pt); % Manualni vytvoreni kontaktu
\draw[fill=white] (con2) circle (1.7pt);
\draw[fill=white] (con7) circle (1.7pt);
\draw[fill=white] (con8) circle (1.7pt);

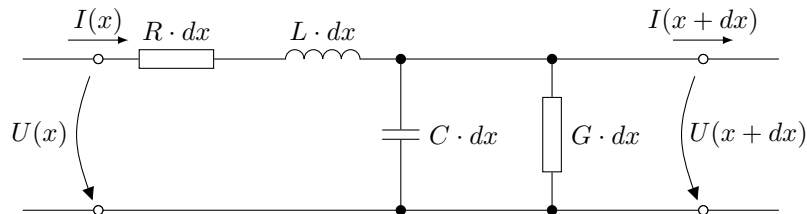
\draw (con1) to [open,v=$U(x)$] (con2); % Napeti U(x)
\draw (con7) to [open,v=$$] (con8); Napeti U(x+dx)
\node at (9.6,1) {$U(x+dx)$};

\draw[->,>=latex] (con1) ++(up:0.25) ++(left:0.4) -- % Proud I(x)
node[left] {} ++(right:0.8);
\node at (1,2.5) {$I(x)$};

\draw[->,>=latex] (con7) ++(up:0.25) ++(left:0.4) -- % Proud I(x+dx)
node[left] {} ++(right:0.8);
\node at (9,2.5) {$I(x+dx)$};
\end{tikzpicture}

```

Výpis 3.3: Náhradní schéma vedení v Circuits



Obrázek 3.4: Náhradní schéma vedení v Circuits

```

\begin{tikzpicture}[circuit ee IEC]
  \node (con1) [contact] at (3,2) {}; % uzel con1 na (3,2)
  \node (con2) [contact] at (3,0) {}; % uzel con2 na (3,0)

  \draw (con2) to [short] (0,0) % kaskada con2-Uin-Zn-con1-Zn-con2
    to [voltage source={direction info={<-}, info={\$U_{in}\$}}] (0,2)
    to [resistor={info=\$Z_n\$}] (con1)
    to [short] (4,2)
    to [resistor={info=\$Z_n\$}] (4,0)
    to (con2);

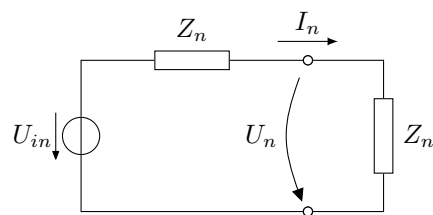
  \draw[fill=white] (con1) circle (1.7pt); % Manualni vytvoreni kontaktu
  \draw[fill=white] (con2) circle (1.7pt);

  \draw (con1) to [open,v=\$U_n\$] (con2); % napeti Un

  \draw[->,>=latex] (con1) ++(up:0.25) ++(left:0.4) -- % proud In
    node[left] {} ++(right:0.8);
  \node at (3,2.5) {\$I_n\$};
\end{tikzpicture}

```

Výpis 3.4: Schéma normálového generátoru v Circuits



Obrázek 3.5: Schéma normálového generátoru v Circuits

ElTeX při sázení funguje v absolutních souřadnicích, tzn. že na začátku definujeme velikost souřadnicové mapy `\begin{picture}(110,50)(0,0)` a následně do ní příkazem `\put(X,Y){PRVEK}` vkládáme jednotlivé prvky. Začátkem každého prvku jsou písmena **h** a **v**, které definují horizontální nebo vertikální otočení součástky.

### 3.4 Circ

Circ je makro pro sazbu základních elektronických schémat. Dostupné je ve verzi 1.0f z roku 2003 a autory jsou Sebastian Tannert a Andreas Tille. Pro jeho funkčnost je potřeba METAFONT a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. Pomocí METAFONT můžeme také makro obohatit o prvky, které nám budou při sazbě chybět.[6]

Circ kromě toho že podporuje většinu základních elektronických a logických prvků, může jako jeden z mála sázet i optické prvky (lasery, čočky, zrcadla, polarizační prvky, atd.)

#### 3.4.1 Instalace

Pro snadnou instalaci makra Circ a správu již instalovaných maker a balíčků doporučuji používat již zmíněný MikTeX. Pokud však MikTeX nemáme, postupujeme podle následujících kroků:

Nejprve si stáhneme adresář s potřebnými soubory a dokumentací makra<sup>1</sup>

Spuštěním souboru `circ.ins` v příkazové řádce L<sup>A</sup>T<sub>E</sub>X dostaneme několik `*.mf` souborů. Tyto soubory pomocí METAFONTu převedeme postupně do fontu `*.pk` a `*.tfm` za pomoci příkazů:

```
mf \mode=localfont; input ciwidko.mf
gftopk ciwidko.300
```

Výsledné soubory `*.pk` a `*.tfm` zkopírujeme společně se soubory `*.sty` a `*.def`, získanými z `circ.ins` do příslušné pracovní složky. Makro by mělo nyní fungovat.

#### 3.4.2 Práce s makrem

Pokud jsme makro úspěšně nainstalovali, zbývá nám jej vložit do hlavičky dokumentu

```
\usepackage[basic,box,gate,oldgate,ic,optics,physics]{circ}
```

Makro pracuje v relativních pozicích, tzn. že při sazbě se pozice vloženého prvku odvíjí od pozice prvku předchozího. Nemusíme tedy řešit přesné rozložení a velikosti mřížky jako např. u makra Eltex. Má poměrně jednoduchou syntaxi založenou na postupném skládání prvků za sebe

---

<sup>1</sup><https://ctan.org/tex-archive/macros/generic/diagrams/circ>

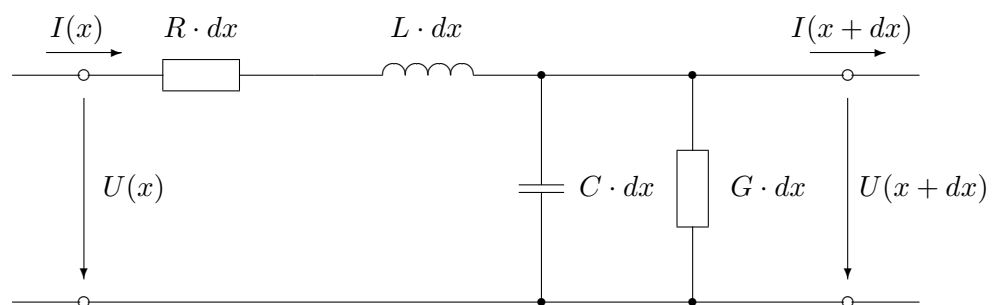


```

\begin{picture}(120,50)(0,0)
\grid{12}{5} % vlozeni mridky pro lepsi orientaci
\put(0,40){\hwire{9}}
\put(0,10){\hwire{9}}
\put(9.5,10){\pin}
\put(9.5,40){\pin}
\put(10,40){\hrez{0}} % vytvoreni horizontalni civky otocene nahoru (U)
\put(40,40){\hind{U}{0}}
\put(70,40){\node} % vlozeni uzlu
\put(70,10){\node}
\put(90,40){\node}
\put(90,10){\node}
\put(70,10){\vcap{0}}
\put(90,10){\vrez{0}}
\put(70,40){\hwire{40}} % vytvoreni horizontalniho vodice o delce 30
\put(10,10){\hwire{100}}
\put(110.5,10){\pin}
\put(110.5,40){\pin}
\put(111,40){\hwire{9}}
\put(111,10){\hwire{9}}
\put(9.5,37){\vector(0,-1){24}} % Napeti U(x)
\put(12,24){$U(x)$}
\put(110.5,37){\vector(0,-1){24}}
\put(112,24){$U(x+dx)$}
\put(4.5,43){\vector(1,0){10}} % Proud I(x)
\put(5,45){$I(x)$}
\put(105.5,43){\vector(1,0){10}}
\put(103,45){$I(x+dx)$}
\put(20,45){$R \cdot dx$} % doplneni popisku k prvku
\put(50,45){$L \cdot dx$}
\put(75,24){$C \cdot dx$}
\put(95,24){$G \cdot dx$}
\end{picture}

```

Výpis 3.5: Náhradní schéma vedení v elTeX



Obrázek 3.6: Náhradní schéma vedení v elTeX

s případným odkazováním na jiné uzly, centrování jednotlivých prvků, atd. Každá elektronická součástka má v dokumentaci zapsán svůj syntax s kódem označujícím počet a formát parametrů.

### 3.5 pst-circ

Pst-circ je makro pro sazbu elektronických obvodů, které je založeno na balíčku `pstricks`, pracujícím s PostScriptem. PostScript je programovací jazyk určený ke grafickému popisu tisknutelných dokumentů vyvinutý v roce 1985 firmou Adobe. Spolu s formátem PDF patří mezi nejpoužívanější formáty pro tvorbu vektorových dat připravených pro tisk. Oproti PDF má však poměrně větší velikost souborů a je náročnější na hardwarové zpracování, díky čemuž se v dnešní době už tolik nepoužívá.[8]

V tomto makru můžeme vytvořit veškeré elektronické prvky, ale také logické komponenty včetně digitálního displeje nebo vlastních integrovaných obvodů s libovolným počtem pinů. K dispozici jsou také mikrovlnné prvky, jako např. oscilátory, filtry, frekvenční násobičky/děličky, fázové posuny, atd.

#### 3.5.1 Instalace

Pro instalaci si musíme nejprve makro stáhnout ze CTAN stránek<sup>1</sup>. Stažené makro jako takové není potřeba instalovat, z archivu pouze zkopírujeme soubory do adresáře nainstalované verze L<sup>A</sup>T<sub>E</sub>X (TeXLive, Miktex, aj.) a to `pst-circ.sty` do složky `./tex/latex/pstricks/` a soubor `pst-circ.pro` do `./dvips/pstricks/`. Ke spuštění jsou dále potřeba balíčky:

`pst-node`, `multido`, `pst-xkey` a samozřejmě nejdůležitější `pstricks`

Všechny tyto balíčky by měly být součástí základní TeXLive nebo Miktex instalace, pokud tomu tak není, stáhneme si chybějící balíčky ze stránek CTAN<sup>2</sup>. Pokud používáme Miktex, může se stát že kód nepůjde v PDFLaTeXu zkompileovat. V takovém případě problém odstraníme kompilací v XeLaTeXu, kde vše funguje.

#### 3.5.2 Práce s makrem

Nainstalované makro vložíme do hlavičky dokumentu společně s knihovnou `pstricks` příkazy:

```
\usepackage{pstricks}
\usepackage{pst-circ}
```

---

<sup>1</sup><https://ctan.org/pkg/pst-circ>

<sup>2</sup><https://ctan.org/>

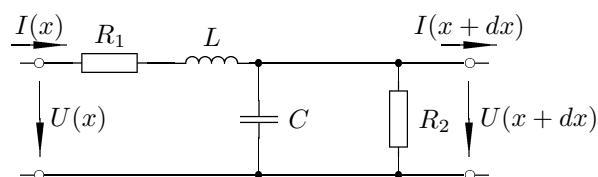
```

\begin{circuit}0
\connection1 {} c r \shift -0.7 0 % vytvoreni kontaktu1 s absolutnim posunem
\ - 2 r % vytvoreni vodice dlouheho 1 s orientaci vpravo
\R1 {} r % vytvoreni rezistoru R1 s orientaci vpravo
\ - 2 r
\L{} {} . r % sazba civky L s orientaci vpravo
\ - 1 r
\ .1 % vytvoreni uzlu ke kteremu se muzeme odkazovat
\ - 2 d
\C{} {} d % vytvoreni C s orientaci dolu
\ - 2 d \ .2 \ - 8 r \ .3 \ - 1 u
\R2 {} u % vytvoreni R2 s orientaci nahoru
\ - 1 u \ .4
\frompin .1 \shift -0.5 0 % presun na uzel 1 s absolutnim posunem
\ - 12 r \shift -0.7 0
\connection3 {} o r
\frompin .3
\ - 4 r \shift -0.7 0
\connection4 {} o r
\frompin .2
\ - 12 l \shift 0.7 0
\connection2 {} o l

\atpin connection1: \shift -2.5 1 \P1 % vytvoreni souradnic pro I(x)
\atpin connection1: \shift 2.5 1 \P2
\Utext{\small $I(x)$} from P1 to P2 % sazba proudu I(x)
\atpin connection3: \shift -2.5 1 \P3 % vytvoreni souradnic pro I(x+dx)
\atpin connection3: \shift 2.5 1 \P4
\Utext{\small $I(x+dx)$} from P3 to P4 % sazba proudu I(x+dx)
\atpin .1: \shift -11.8 0 \P3 % vytvoreni souradnic pro napeti U(x)
\atpin .2: \shift -11.8 0 \P4
\Utext{\small $U(x)$} from P3 to P4 % napeti U(x)
\atpin .4: \shift 3.8 0 \P3 % vytvoreni souradnic pro napeti U(x+dx)
\atpin .3: \shift 3.8 0 \P4
\Utext{\small $U(x+dx)$} from P3 to P4 % napeti U(x+dx)
\end{circuit}

```

Výpis 3.6: Náhradní schéma vedení v makru Circ



Obrázek 3.7: Náhradní schéma vedení v makru Circ

```

\begin{circuit}0
\connection1 {} o l % vytvoreni kontaktu1
\ - 3 l % vytvoreni vodice dlouheho 3 s orientaci vlevo
\R1 {} l
\ - 3 l
\ - 2 d
\cc\varU1 {} d % sazba zdroje U1 s otocnym popiskem
\ - 2 d
\ - 10 r
\connection2 {} c r \shift -0.3 0 % vytvoreni kontaktu2 s absolutnim posunem
\shift -0.4 0
\ - 5 r
\ - 2 u
\R2 {} u
\ - 2 u
\ - 5 l

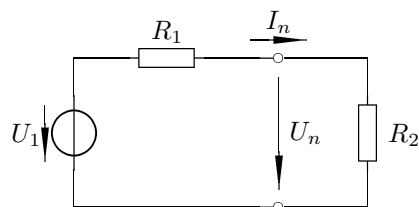
\atpin connection1: \shift -2.5 1 \P1 % vytvoreni souradnic pro proud In
\atpin connection1: \shift 2.5 1 \P2
\Utext{\small  $I_n$ } from P1 to P2 % sazba proudu In

\atpin connection1: \P3 % vytvoreni souradnic pro napeti U(n)
\atpin connection2: \P4
\Utext{\small  $U_n$ } from P3 to P4 % napeti U(n)

\atpin varU1: \shift -1.6 -2.5 \P3 % vytvoreni souradnic pro napeti U1
\atpin varU1: \shift -1.6 2.5 \P4
\Utext{} from P4 to P3 % sazba napeti U1
\end{circuit}

```

Výpis 3.7: Schéma normálového generátoru v makru Circ



Obrázek 3.8: Schéma normálového generátoru v makru Circ

S makrem pracujeme v PostScriptovém prostředí `pspicture`, kterému je nutné definovat velikost pracovní mřížky. Podobně jako `Eltex19` pracuje makro v absolutních souřadnicích, proto je při sazbě výhodné použít pro snadnou orientaci číslovanou mřížku:

```
\begin{pspicture}[showgrid=true](0,0)(11,5)

\end{pspicture}
```

Výpis 3.8: Prostředí `pspicture` s velikostí mřížky 11,5

V prostředí `pspicture` už můžeme vytvářet jednotlivé elektronické prvky. Nejprve je výhodné definovat si množinu uzlů (Pokud je budeme potřebovat), kterým přiřadíme označení a souřadnice. Následně se můžeme k uzlům odkazovat a používat jejich názvy jako souřadnic pro rozmísťování součástek nebo vodičů.

### 3.6 Latex Circuit Diagram

Latex Circuit Diagram (dále LCD) je velice jednoduché makro vytvořené roku 2011 Kimmo Silvonenem a Vesa Linja-aho z finské Aalto University, ELEC (School of Electrical Engineering).[10] Makro je postaveno na prostředí `picture`, které je součástí instalace  $\text{\LaTeX}$  2 $\epsilon$  a umožňuje snadno vytvářet základní obrazce složené z čar, čtyřúhelníků, elips, bézierových křivek nebo také textu.[12] Díky použití základního prostředí je makro použitelné jak na platformě Windows, tak i Linux.

LCD můžeme stáhnout ze stránek GoogleCode<sup>3</sup> v ZIP souboru obsahujícím  $\text{\TeX}$  soubory `latex-macro`, který definuje jednotlivé komponenty jako nové příkazy v prostředí `picture`, `list_of_macros` a PDF soubor, který je kompilací souboru `list_of_macros` a zobrazuje rejstřík všech komponent včetně příkladů jednoduchých zapojení. Pro zprovoznění makra stačí zkopírovat soubor `latex-macro` do adresáře projektu a poté jej nahrát do hlavičky našeho dokumentu příkazem:

```
\input{latex-macro.tex}
```

Dále musíme nastavit v těle dokumentu velikost jednotek v prostředí `picture`. Jde o jakési poměrné měřítko mezi bodem v prostředí `picture` a bodem ve výstupním dokumentu. Velikost můžeme vyjádřit v milimetrech, bodech (*pt* - jeden bod je 1/72 palce), pixelech (*px*) nebo "em" (*em* - jednotka měnící svou velikost v závislosti na aktuální velikosti písma. 1em = výška jednoho řádku základního písma)[9]. Za předpokladu že používáme pro sazbu standardní dokument A4 a jelikož základní RLC součástka má v nadefinovaném makru velikost 50 jednotek, doporučuji používat pro menší schémata velikost jednotek prostředí přibližně 1pt. Zápis příkazu pak vypadá následovně:

---

<sup>3</sup><https://code.google.com/archive/p/latex-circuit-diagram/>

```

\begin{pspicture}[showgrid=true] (0,0) (14,5)
\pnodes(1,4){A}(1,1){B}(2,4){C}(2,1){D}(7,4){E}(7,1){F}(10,4){G}(10,1){H}(12,4)
  {I}(12,1){J}(13,4){K}(13,1){L}(2,3.9){ux1}(2,1.1){ux2}(12,3.9){uxdx
  1}(12,1.1){uxdx2}(1.6,4.3){ix1}(2.4,4.3){ix2}(11.6,4.3){idx1}(12.4,4.3){
  idx2} % Definovani uzlu

\qdisk(E){2pt}\qdisk(F){2pt}\qdisk(G){2pt}\qdisk(H){2pt} % Vyznaceni uzlu
\multidipole(C)(E)\resistor{$R \cdot dx$}
\coil{$L \cdot dx$}. % Kaskada R - L

\resistor[labeloffset=1](G)(H){$G \cdot dx$} % G
\capacitor[labeloffset=1](E)(F){$C \cdot dx$} % C

\tension[labeloffset=-0.5](ux1)(ux2){$U(x)$} % Napeti U(x)
\tension[labeloffset=0.8](uxdx1)(uxdx2){$U(x+dx)$} % Napeti U(x+dx)

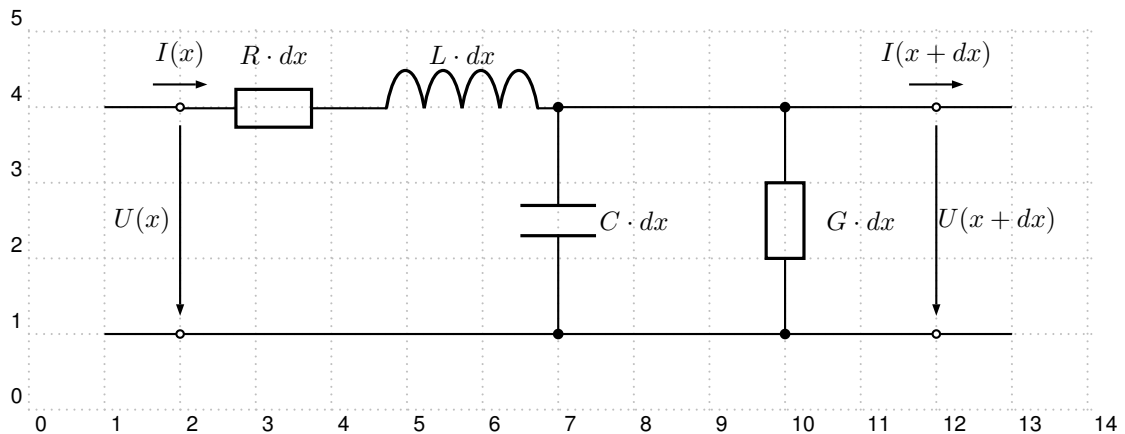
\tension[labeloffset=0.4](ix1)(ix2){$I(x)$} % Proud I(x)
\tension[labeloffset=0.4](idx1)(idx2){$I(x+dx)$} % Proud I(x+dx)

\wire(D)(12,1) % Propojovací vodice
\wire(E)(12,4)

\wire[arrows=->](A)(C) % Kontakty
\wire[arrows=->](B)(D)
\wire[arrows=->](I)(K)
\wire[arrows=->](J)(L)
\end{pspicture}

```

Výpis 3.9: Náhradní schéma vedení v makru pst-circ



Obrázek 3.9: Náhradní schéma vedení v makru pst-circ

```
\setlength{\unitlength}{1pt}
```

Makro pracuje podobně jako Eltex19 nebo pst-circ26 v absolutních souřadnicích. Musíme tedy nejprve vytvořit prostředí o určité velikosti a do něj poté dosazovat jednotlivé prvky v absolutních souřadnicích. Nulový bod prvku, neboli bod který se dosazuje na zadané souřadnice, se nachází v závislosti jestli je součástka horizontální - *vlevo* nebo vertikální - *dole*.

### 3.7 CircDia

CircDia<sup>4</sup> je poměrně nové makro vytvořené Stefanem Krausem a publikované 24. února 2018. V CircDia se stejně jako v LCD makru pracuje v absolutních pozicích a je taktéž dostupné jak pro platformu Windows, tak i Linux. Oproti LCD je však postaveno na balíčku `tikz` a nabízí daleko více komponent s různými variantami popisu a hodnotou součástky.

Instalace probíhá stažením ZIP souboru s potřebnými soubory na domovské stránce balíčku, rozbalením, zkopírováním složky `circdia` do adresáře `./tex/latex/` a spuštěním příkazu `texhash` pokud používáme TeXLive a u Miktexu v okně nastavení stisknutím tlačítka **Refresh FNDB**. Do našeho dokumentu potom makro nahrajeme pomocí příkazu:

```
\usepackage{circdia}
```

Pro sazbu si stáhneme opět z domovské stránky makra PDF dokument<sup>5</sup>, který je prozatím v němčině, nicméně pro základní pochopení je plně dostačující a autor plánuje v nejbližší době také anglický překlad. V dokumentu najdeme přehledně popsané všechny prvky, které makro nabízí a jejich zápis v systému L<sup>A</sup>T<sub>E</sub>X. Stejně jako u LCD makra začíná sazba vytvořením prostředí pro sazbu `circuitdiagram` a nadefinováním jeho velikosti. Do tohoto prostředí potom sázíme jednotlivé prvky podle jejich typu, souřadnice, orientace, názvu a také hodnoty. Oproti LCD se makro CircDia liší umístěním nulových bodů uprostřed a různými velikostmi prvků, což je při dosazování vodičů nebo uzlů potřeba řešit metodou pokus omyl.

---

<sup>4</sup><http://www.taylorgruppe.de/circdia/>

<sup>5</sup><http://www.taylorgruppe.de/circdia/circdia-doc-de.pdf>

```

\begin{picture}(200,50)(0,0) % Prostredi o velikosti 200x50 s 0,0 offsetem
\hln{0,50}{20}
\out{20,0} % Kontakty
\out{20,50}
\out{180,50}
\out{180,0}

\hz{20,50}{R \cdot dx} % Horizontalni a vertikalni prvky
\hl{60,50}{L \cdot dx}
\vc{110,0}{C \cdot dx}
\vz{160,0}{G \cdot dx}

\hln{100,50}{100} % Horizontalni cara o delce 100 na 100,50
\hln{0,0}{200}

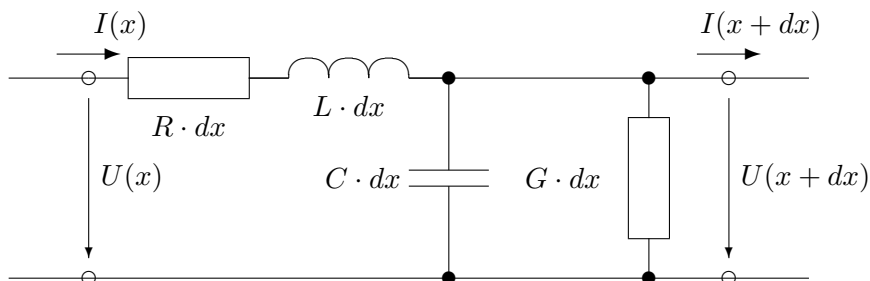
\du{20,0}{U(x)} % Napeti U(x) a U(x+dx)
\du{180,0}{U(x+dx)}

\rui{28,56}{I(x)} % Proud I(x) a I(x+dx)
\hln{12,56}{13}
\rui{188,56}{I(x+dx)}
\hln{172,56}{13}

\cn{110,0} % Vytvoreni uzlu
\cn{110,50}
\cn{160,0}
\cn{160,50}
\end{picture}

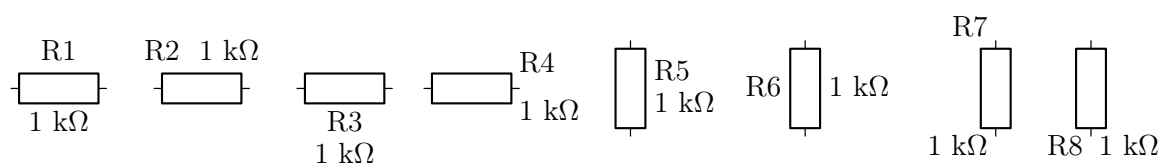
```

Výpis 3.10: Náhradní schéma vedení v makru Latex Circuit Diagram



Obrázek 3.10: Náhradní schéma vedení vytvořeného pomocí Latex Circuit Diagram





```
\begin{circuitdiagram}{72}{10}
\resis{3}{5}{H}{R1}{1 k\Ohm}
\resis{12}{5}{Hu}{R2}{1 k\Ohm}
\resis{21}{5}{Hdd}{R3}{1 k\Ohm}
\resis{29}{5}{Hr}{R4}{1 k\Ohm}
\resis{39}{5}{V}{R5}{1 k\Ohm}
\resis{50}{5}{Vlr}{R6}{1 k\Ohm}
\resis{62}{5}{Vhl}{R7}{1 k\Ohm}
\resis{68}{5}{Vd}{R8}{1 k\Ohm}
\end{circuitdiagram}
```

Obrázek 3.11: Možnosti různého uspořádání textu součástek u makra CircDia

```

\begin{circuitdiagram}[40]{10}
\wire{0}{10}{5}{10}
\resis{8}{10}{H}{\$R\cdot dx\$}{} % Serie R - L
\wire{11}{10}{13}{10}
\induc{16}{10}{H}{\$L \cdot dx\$}{}
\wire{19}{10}{41}{10}
\capac{21}{5}{V}{\$C \cdot dx\$}{} % Vertikalni C
\resis{30}{5}{V}{\$G \cdot dx\$}{} % Vertikalni R
\wire{21}{10}{21}{6} % Propojovací vodice
\wire{21}{4}{21}{0}
\wire{30}{10}{30}{8}
\wire{30}{2}{30}{0}
\wire{0}{0}{41}{0}

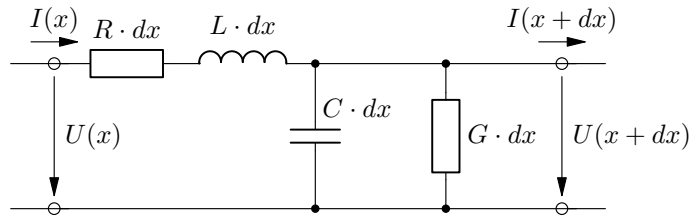
\junct{21}{10}      % Pridani 4 uzlu
\junct{30}{10}
\junct{21}{0}
\junct{30}{0}

\pin{3}{10}{L}{}    % Kontakty
\pin{3}{0}{L}{}
\pin{38}{10}{L}{}
\pin{38}{0}{L}{}

\Voltarrow{3}{10}{3}{0}{r}{\$U(x)\$} % Napeti U(x) a U(x+dx)
\Voltarrow{38}{10}{38}{0}{r}{\$U(x+dx)\$}
\Voltarrow{0.5}{11.5}{5.5}{11.5}{u}{\$I(x)\$} % Proud I(x) a I(x+dx)
\Voltarrow{35.5}{11.5}{40.5}{11.5}{u}{\$I(x+dx)\$}
\end{circuitdiagram}

```

Výpis 3.11: Náhradní schéma vedení v makru CircDia



Obrázek 3.12: Náhradní schéma vedení vytvořeného pomocí balíčku CircDia

## 4 WYSIWYG editory

WYSIWYG je akronym z anglického "What You See Is What You Get", což můžeme česky volně přeložit jako "co vidíš, to dostaneš". Tato zkratka označuje způsob editace dokumentů, při kterém je verze zobrazená na obrazovce vzhledově totožná s výslednou verzí dokumentu. Nejznámějším zástupcem textových WYSIWYG editorů je bezpochyby MS Word nebo OpenOffice.[13]

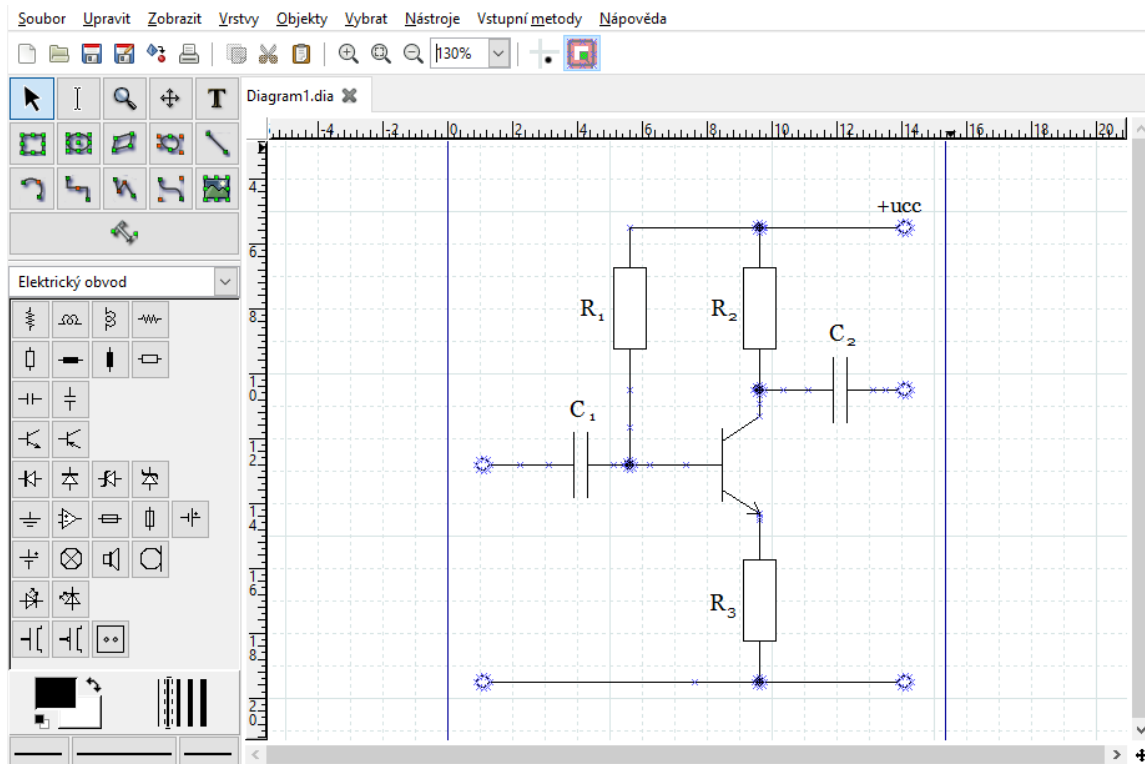
Účelem takových editorů je maximálně zrychlit a zjednodušit vytvoření jednotlivých schémat s minimální nebo žádnou znalostí některého z balíčků pro sazbu. Princip sazby většinou vypadá tak, že si pomocí grafického rozhraní editoru volíme prvky, které následně rozmisťujeme do pracovní plochy editoru a tím postupně vytváříme jednotlivá schémata. Následný export schéma z editoru do samotného L<sup>A</sup>T<sub>E</sub>X dokumentu probíhá buď ve formě T<sub>E</sub>X zdrojového kódu, který se poté přímo zkopíruje nebo externě nahraje do našeho dokumentu nebo ve formě externího vektorového souboru (.EPS, .PDF), který lze externě nahrát do dokumentu. V této práci však budou zmíněny pouze editory, které nabízí export přímo do T<sub>E</sub>X souboru, ať už jako kód určitého balíčku nebo prostředí `tikzpicture` s převedením schéma na elementární prvky.

### 4.1 Dia

Dia je Open Source kreslicí software pro Windows, Linux a Mac OS. Podporuje více než 30 různých typů diagramů, jako jsou vývojové diagramy, CISCO síťové topologie, databázové modely, logické obvody nebo pro nás důležité elektronické schémata. Pro tvorbu elektronických schémat v tomto editoru nemusíme znát způsob sazby schémat pomocí základních knihoven, ani jejich syntaxi.

Na obrázku 4.1 můžeme vidět jak vypadá rozhraní editoru připravené pro sazbu elektronických schémat. Abychom nemuseli elektronické prvky sami vytvářet pomocí základních objektů (což samozřejmě není problém). Je pro nás připravena knihovna se základními prvky, kterým určíme pouze velikost a tvar. Knihovnu načteme kliknutím na roletku na levé straně pod výběrem základních objektů `Ostatní listy -> Elektrický obvod`. Po výběru se nám zobrazí panel nástrojů s ikonkami jednotlivých prvků pro horizontální a vertikální sázení objektů, jelikož editor bohužel nepodporuje rotaci objektů. Vpravo vidíme pracovní plochu pro sázení prvků. Pro lepší orientaci si můžeme zapnout v záložce `Zobrazit` mřížku a přichytávání objektů k ní.

Exportovat z programu můžeme do různých rastrových a vektorových formátů, ale hlavně také do T<sub>E</sub>Xsouboru, kdy nám editor vygeneruje `tikzpicture`, který můžeme snadno přímo nebo externě vložit do souboru s dokumentem. Generovaný kód však může zabírat i dvacetinásobek řádků kódu než při použití základních knihoven, např. `Circuitikz` ?? . Měli bychom proto používat Dia editor jen pro sazbu jednodušších schémat a pro složitější zvolit vhodnější způsob.

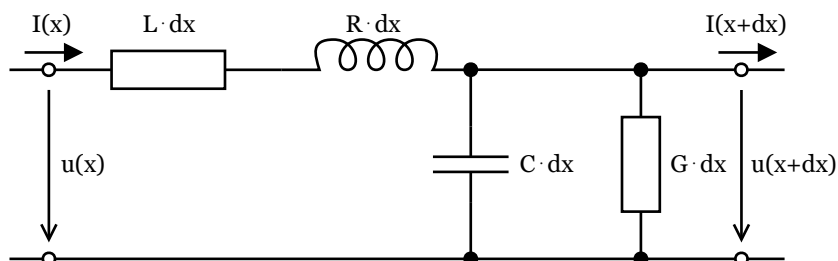


Obrázek 4.1: Rozhraní editoru Dia

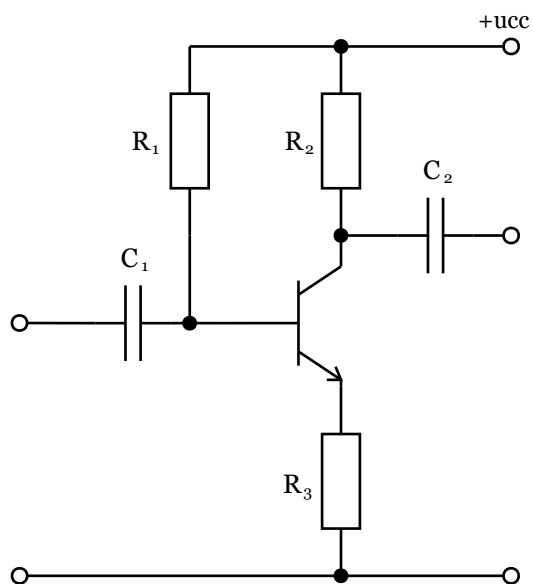
## 4.2 TikzEdt

Trochu rozdílným editorem oproti Dia je TikzEdt<sup>6</sup>, který je WYSIWYG/textový, což znamená že můžeme upravovat schéma v kódu a v reálném čase uvidíme výsledek na náhledové pracovní ploše nebo můžeme naopak upravovat na pracovní ploše a v kódu se nám budou měnit hodnoty. TikzEdt je dostupný pro platformu Windows nebo Linux v beta verzi 2.3.0 pro Windows a 2.2.0 pro Linux (V nižších verzích neobsahuje knihovny se základními elektronickými prvky).

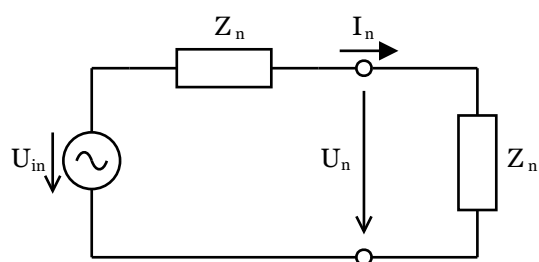
<sup>6</sup><http://www.tikzedt.org/>



Obrázek 4.2: Náhradní schéma vedení kreslené v Dia



Obrázek 4.3: Schéma tranzistorového zesilovače kreslené v Dia

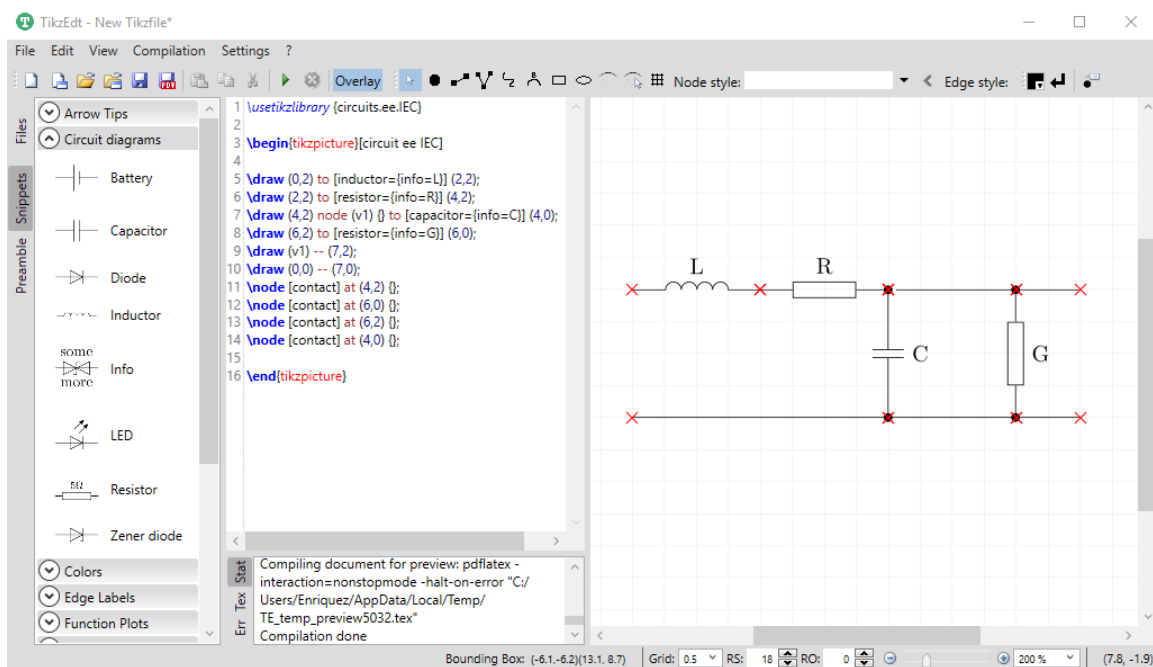


Obrázek 4.4: Schéma normálového generátoru kreslené v Dia

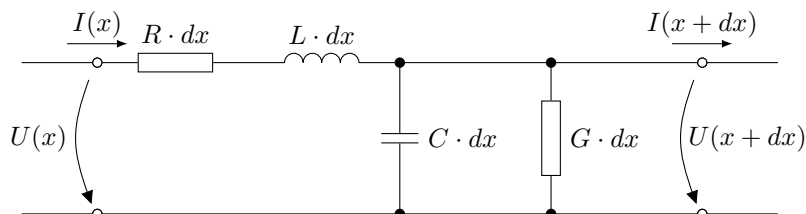
Editor pro obě platformy stáhneme z GoogleCode v sekci download. Program jako takový není potřeba instalovat, pro Windows stáhneme ZIP soubor, jehož archiv pouze extrahujeme a můžeme spustit program. Pro Linux je TikzEdt dostupný pro 32 bitovou a 64 bitovou architekturu. Po stažení spustíme ve složce kde je soubor uložen pro extrahování příkaz: `tar -xf <Název souboru>` a poté můžeme spustit aplikaci `./tikzedt`.

Samotné sázení probíhá trochu nešťastně. Editor sice má knihovnu s elektronickými prvky, nicméně jedná se pouze o makra, které vykreslí základní prvek bez souřadnic a parametrů. Musíme proto ovládat alespoň základy knihovny `Circuits`, kterou editor ze základu podporuje, abychom chybějící údaje doplnili a upravili syntaxi mezi jednotlivými prvky. Práce v programu tedy vypadá tak, že v textovém okně vkládáme prvky z knihoven v levém sloupečku nebo přímo textově a v náhledovém okně můžeme tyto prvky poté pozicovat pomocí červených křížků znázorňujících uzly elementů nebo vytvářet jednoduché tvary jako jsou body, křivky znázorňující vodiče a základní geometrické tvary. U tohoto editoru je doporučeno používat instalaci *TeXLive*, jelikož u instalace MiKTeX může nastat problém při kompilaci.

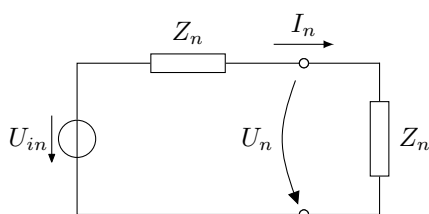
Pro pokročilé uživatele existuje také možnost používat jiný balíček než `Circuits` a to upravením hlavičky dokumentu v záložce `Settings > Settings... > Compiler`. Můžeme také pro daný balíček upravit knihovny se základními prvky opět v záložce `Settings > Snippet Manager...`, kde po rozkliknutí komponenty lze v poli `Sample Code` editovat kód, který součástka po přidání vypíše.



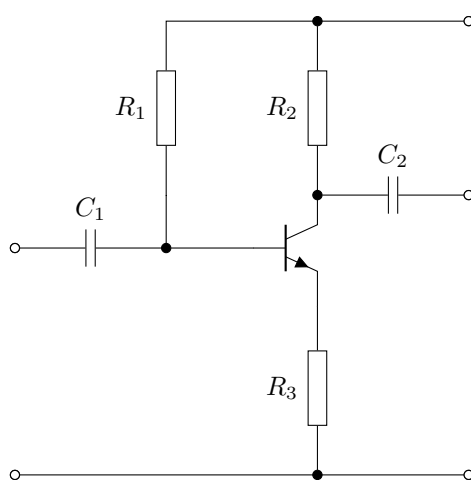
Obrázek 4.5: Práce v editoru TikzEdt



Obrázek 4.6: Náhradní schéma vedení vytvořené v TikzEdt s knihovnou Circuits



Obrázek 4.7: Schéma normálového generátoru vytvořené v TikzEdt s knihovnou Circuits



Obrázek 4.8: Schéma tranzistorového zesilovače vytvořené v TikzEdt s knihovnou Circuits

### 4.3 Inkscape

Inkscape<sup>1</sup> je open-source vektorový grafický editor podobný profesionálním programům Adobe Illustrator nebo CorelDraw. Jako jediný používá soubory .SVG jako svůj nativní formát.[14] Práce na Inkscape začala roku 2003 jako sesterský projekt vektorového editoru Sodipodi. V současné době je dostupný pro platformy Linux, Windows a Mac OS X ve verzi 0.92. Jako jeden z mála editorů nabízí export grafiky přímo do  $\text{\TeX}$  souboru pomocí PostScriptového rozhraní `pspicture`, ale samozřejmě také do PDF nebo EPS.

Pro vytváření schémat v editoru Inkscape existuje několik uživateli vytvořených rozšíření, které by měli urychlit práci. Nejzajímavější se jeví rozšíření `CircuitSymbols`<sup>2</sup>, které umožňuje po přidání několika souborů do složky `extensions` v adresáři instalace, vkládat jednotlivé prvky přímo do pracovní plochy editoru přes okno, kde nastavíme jakou součástku chceme vložit, její hodnotu, orientaci a další parametry a tlačítkem **použít** ji přidáme do pracovní plochy. Toto rozšíření bylo vytvořeno primárně pro Inkscape 0.48 a částečně 0.91 v systému Linux (Kubuntu 12.04 a 14.04), podle slov autora by však mělo fungovat i pod systémem Windows. Nicméně při vší snaze se mi rozšíření ani na systému Linux ani Windows x64 ani x32 nepodařilo plně zprovoznit.

Další možností tvorby schémat, jelikož Inkscape pracuje s formátem .SVG, je stažení .SVG kolekce s elektrickými symboly například z webu [commons.wikimedia.org](https://commons.wikimedia.org)<sup>3</sup>. Kdy si stažený soubor otevřeme v programu Inkscape a z jednotlivých objektů vytváříme schéma, kromě vodičů, které vytvoříme pomocí nástroje **Kresba Bézierových křivek a přímek**. V roletce **Zobrazení** -> **Mřížka stránky** nebo klávesou **#** si můžeme zapnout mřížku, ke které se budou objekty automaticky přichytávat.

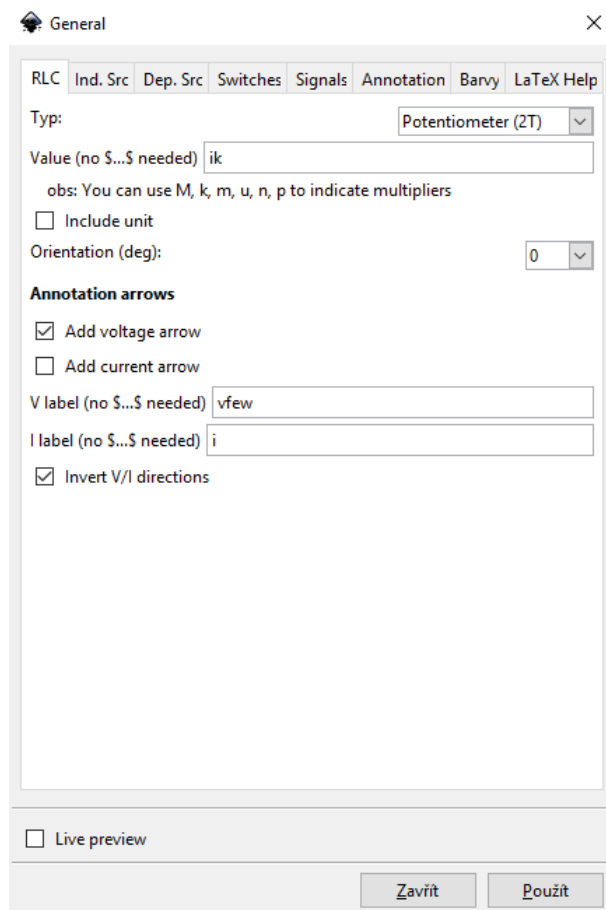
---

<sup>1</sup><https://inkscape.org/en/release/0.92.3/>

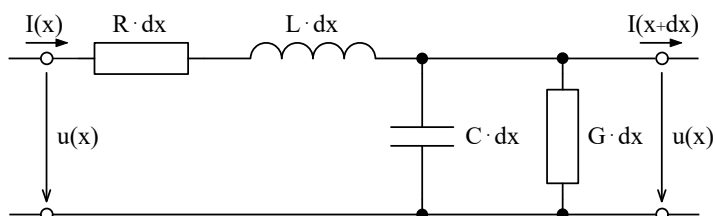
<sup>2</sup><https://github.com/fsmMLK/inkscapeCircuitSymbols>

<sup>3</sup>[https://commons.wikimedia.org/wiki/File:Electrical\\_Symbols\\_IEC.svg](https://commons.wikimedia.org/wiki/File:Electrical_Symbols_IEC.svg)

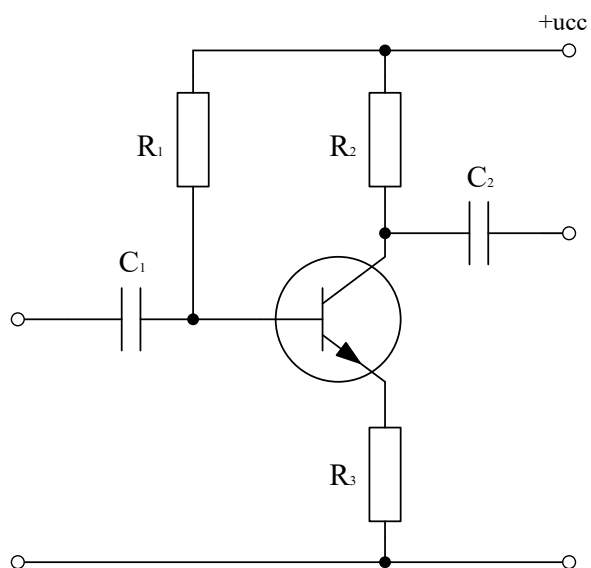




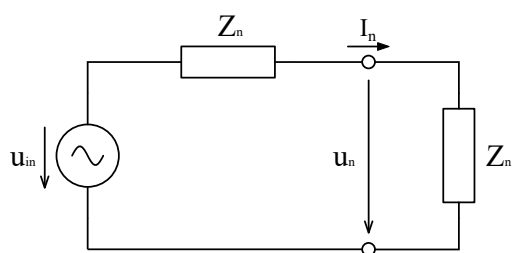
Obrázek 4.9: Okno rozšíření CircuitSymbols pro nastavení a vložení RLC prvků



Obrázek 4.10: Náhradní schéma vedení kreslené v Inkscape



Obrázek 4.11: Schéma tranzistorového zesilovače kreslené v Inkscape



Obrázek 4.12: Schéma normálového generátoru kreslené v Inkscape

## 5 Návod pro práci s balíčky

### 5.1 Circuitikz

Jelikož patří Circuitikz mezi nejrozšířenější a snadno pochopitelné balíčky s velkou knihovnou použitelných prvků, podrobně si jej rozebereme a vysvětlíme. Pro práci s balíčkem je také vhodné používat oficiální manuál s přehledem všech dostupných prvků a jejich zápisem. Manuál je dostupný online na domovské CTAN stránce.<sup>7</sup>

Jak již bylo zmíněno, abychom mohli balíček používat, musíme jej nejprve definovat v hlavičce dokumentu příkazem:

```
\usepackage{circuitikz}
```

Circuitikz ve výchozím nastavení používá při sazbě americké komponenty, toto nastavení však můžeme změnit použitím jednoho nebo více parametrů v definici knihovny. Pro nás bude výhodné použít parametr *european*, který obsahuje množinu evropských prvků jako *europeanresistors*, *europeaninductors*, *europeancurrents*, atd. Pokud chceme používat jednotky jako  $\Omega$ ,  $\mu\text{F}$  nebo H, je nutné definovat knihovnu *siunitx*.

```
\usepackage[european, siunitx]{circuitikz}
```

Samotný kód píšeme v těle dokumentu v prostředí *circuitikz*, kterému můžeme nastavit vlastnosti jako celkovou velikost vykresleného schéma nebo tloušťku čar.

```
\begin{circuitikz}
...
Circuitikz kod
...
\end{circuitikz}
```

#### 5.1.1 Komponenty

Při sazbě jednotlivých komponent rozlišuje Circuitikz mezi tzv. **Normálními** komponenty, které se chovají jako jeden uzel, kterému stačí nastavit pouze jedny XY souřadnice a **Bipóly**, které vyžadují počáteční souřadnice pro první pól a koncové souřadnice pro druhý pól.

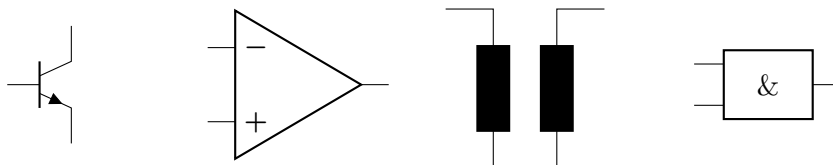
Mezi **Normální komponenty** patří všechny jednopólové nebo více než dvoupólové prvky. Najdeme zde například tranzistory, operační zesilovače, transformátory, ale i logické prvky, atd. Syntax pro tyto komponenty vypadá následovně:

```
\draw (0,0) node[a,b] (c) {d};
```

---

<sup>7</sup><https://ctan.org/pkg/circuitikz>

- (a) Název komponentu (povinné)
- (b) Jeden nebo více parametrů oddělených čárkou (volitelné)
- (c) Název pro pozdější odkazování k prvku (volitelné)
- (d) Popisek prvku (volitelné, ale je nutné alespoň zapsat prázdné složené závorky )



```
\begin{circuitikz}
\draw (0,0) node[npn]{};
\draw (3,0) node[op amp]{};
\draw (6,1) node[transformer]{};
\draw (10,0) node[european and port]{};
\end{circuitikz}
```

Výpis 5.1: Příklady zápisu normálních prvků

**Bipólové komponenty** zahrnují RLC prvky, diody, napájecí zdroje, spínací prvky, atd. Zapisujeme je následovně:

```
\draw (0,0) to [a,b] (2,0);
```

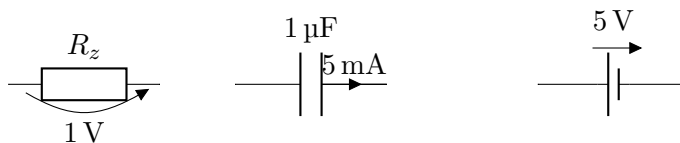
- (a) Název komponentu (povinné)
- (b) Jeden nebo více parametrů oddělených čárkou (volitelné)



```
\begin{circuitikz}
\draw (0,0) to [R] (2,0);
\draw (3,0) to [Di] (5,0);
\draw (6,0) to [european voltage source] (8,0);
\draw (9,0) to [switch] (11,0);
\end{circuitikz}
```

Výpis 5.2: Příklady zápisu bipólových prvků

U každého bipólového prvku můžeme nastavit jeho jméno, popřípadě hodnotu, napětí a proud. Tyto parametry nastavujeme za názvem prvku (*b*) a oddělujeme je čárkou. Pokud chceme použít u názvu složitější formátování, např. dolní index nebo zlomek, je nutné výraz vložit mezi znaky \$.



```
\begin{circuitikz}
\draw (0,0) to[R=$R_z$, v=1<\volt>] (2,0);
\draw (3,0) to[C=1<\micro\farad>, i=5<\milli\ampere>] (5,0);
\draw (7,0) to[battery1=5<\volt>] (5,0);
\end{circuitikz}
```

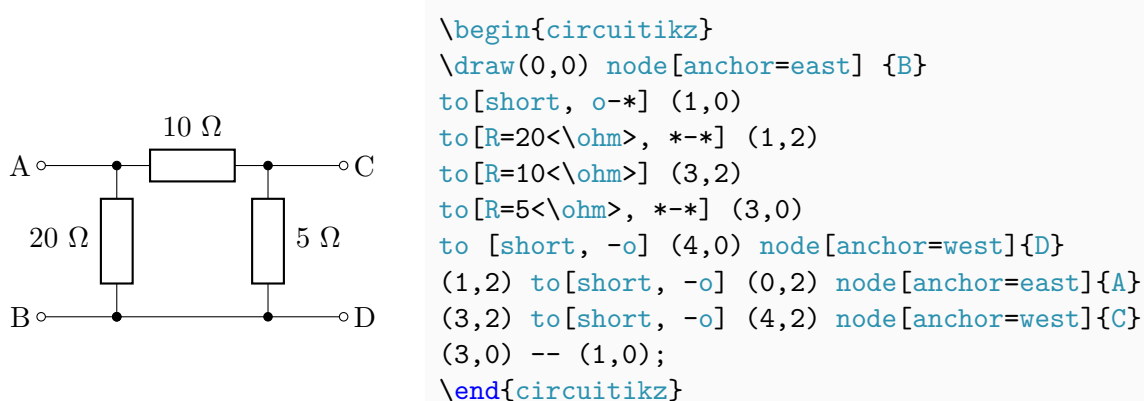
Výpis 5.3: Příklady použití názvů a hodnot prvků

### 5.1.2 Sazba a spojování prvků

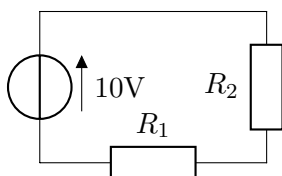
Circuitikz nabízí při sazbě prvků použití absolutních nebo relativních souřadnic viz. 3.1 Pokud použijeme absolutní souřadnice, je výhodné vycházet z levého spodního rohu schéma u souřadnice (0,0). Použití relativních souřadnic je výhodné například při kreslení schéma s vícepólových prvkem uprostřed (tranzistor, zesilovač, atd.).

Pro kreslení prvků použijeme příkaz `\draw`, za kterým následují počáteční souřadnice a následně jeden nebo více prvků oddělených příkazem `to`. S jedním příkazem `\draw` tedy můžeme vytvořit celou kaskádu spojených prvků. Celý příkaz je nutné pro správnou funkci zakončit středníkem.

Pokud chceme spojit dva body pouze vodičem bez použití prvku, můžeme využít znaků `-` mezi dvěma souřadnicemi nebo použít prázdný prvek `[short]`. Při použití relativních souřadnic se příkaz liší pouze přidáním znaků `++` před koncovými souřadnicemi.

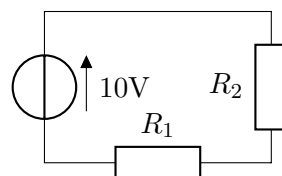


Obrázek 5.1: Použití parametru pro vyznačení uzlů a kontaktů



```
\begin{circuitikz}
\draw (0,0) to [R=$R_1$] (3,0)
to [R=$R_2$] (3,2) -- (0,2)
to [V<=10V] (0,0);
\end{circuitikz}
```

Výpis 5.4: Použití absolutních souřadnic



```
\begin{circuitikz}
\draw (0,0) to [R=$R_1$] ++(3,0)
to [R=$R_2$] ++(0,2) ---+(-3,0)
to [V<=10V] (0,0);
\end{circuitikz}
```

Výpis 5.5: Použití relativních souřadnic

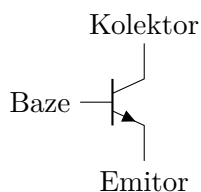
Mezi parametry každého prvku patří také vyznačení počátečního a koncového uzlu `*` pro uzel a `o` pro kontakt. Celý příkaz tedy například vypadá: `o-*`. Viz obrázek 5.1.

Pokud chceme sázet normální vícepólové prvky jako jsou tranzistory, logické prvky nebo například zesilovače. Vytvoříme nejdříve samotný prvek na určitých souřadnicích a pomocí názvu (*c*) se můžeme k prvku odkazovat na jednotlivé piny. Například pokud vytvoříme NPN tranzistor příkazem:

```
\draw (0,0) node[npn] (npn1) {};
```

Můžeme se odkazovat k jeho pinům názvem prvku a názvem pinu odděleného tečkou. Takový příkaz se dá použít jako souřadnice pro další kreslení.

Jak již bylo řečeno, u takových obvodů je doporučeno používat relativní souřadnice, jelikož utrefit přesnou pozici pinu nebo bodu kolmého k pinu je v absolutních souřadnicích obtížné.

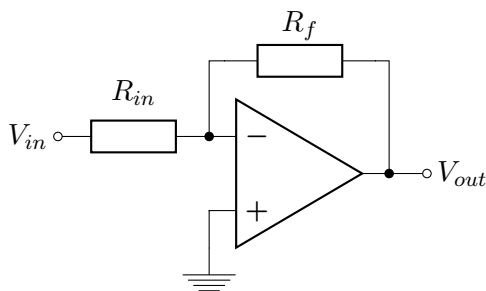


```
\begin{circuitikz}
\draw(0,0) node[npn] (npn) {}
(npn.base) node[anchor=east] {Baze}
(npn.collector) node[anchor=south] {Kolektor}
(npn.emitter) node[anchor=north] {Emitor};
\end{circuitikz}
```

Obrázek 5.2: Vykreslení popsaného NPN tranzistoru

```
\begin{circuitikz}
\draw (0,0) node[op amp] (opamp) {}
(opamp.-) to [R, l_=$R_{in}$, *-o] ($(opamp.-)-(2,0)$) node[left]{$V_{in}$}
(opamp.-) |- ($(opamp.-)+(0.2,1)$) to[R=$R_f$] ($(opamp.-)+(2.2,1)$) -|
(opamp.out) to[short,*-] ($(opamp.out)+(.5,0)$) node[right]{$V_{out}$} node [
ocirc] {}
(opamp.+) to[short] ($(opamp.+)-(0,.5)$) node[ground] {};
\end{circuitikz}
```

Výpis 5.6: Schéma normálového generátoru v CircuiTikz



Obrázek 5.3: Schéma invertujícího zesilovače

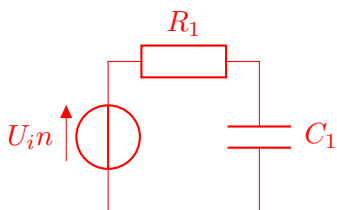
### 5.1.3 Přizpůsobení

Balíček Circuitikz obsahuje řadu klíčových definic, které upravují veškeré grafické vlastnosti knihovny od základních věcí jako je celková velikost schéma, tloušťka čar, barva jednotlivých komponent, až po délku a šířku komponent nebo pozici šipek pro napětí a proud na prvku.

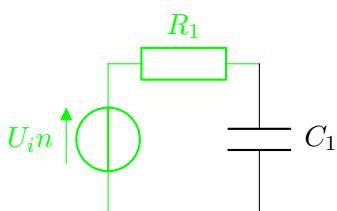
Začneme nastavením velikosti výsledného schéma pomocí příkazu `scale`, který ze základu obsahuje referenční hodnotu 1.0, tedy 100%. Zvýšením hodnoty např. na 1.5 zvětšíme schéma oproti výchozí hodnotě o 150% a naopak.

```
\begin{circuitikz}[scale=1.2]
...
Circuitikz kod
...
\end{circuitikz}
```

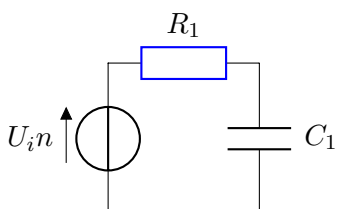
Potřebujeme-li ve schématu vyznačit barevně určitou komponentu nebo část, nabízí Circuitikz většinu základních barev, které můžeme přiřadit jako parametr celému schématu, jednomu příkazu `\draw` nebo určitému prvku.



```
\begin{circuitikz}[color=red]
\draw (0,0) to [V=$U_{in}$] (0,2)
      to [R=$R_1$] (2,2);
\draw (2,2) to [C=$C_1$] (2,0)
      to (2,0) -- (0,0);
\end{circuitikz}
```



```
\begin{circuitikz}
\draw[color=green] (0,0) to [V=$U_{in}$] (0,2)
      to [R=$R_1$] (2,2);
\draw (2,2) to [C=$C_1$] (2,0)
      to (2,0) -- (0,0);
\end{circuitikz}
```



```
\begin{circuitikz}
\draw (0,0) to [V=$U_{in}$] (0,2)
      to [R=$R_1$, color=blue] (2,2);
\draw (2,2) to [C=$C_1$] (2,0)
      to (2,0) -- (0,0);
\end{circuitikz}
```



Pokud chceme změnit tloušťku čar, velikost, proporce nebo vlastnosti určitých prvků, použijeme příkaz `\ctikzset`, do kterého uvedeme množinu prvků nebo jeden určitý prvek a následnou akci která se má vykonat. Příkaz můžeme umístit přímo do kódu prostředí `circuitikz` nebo do těla dokumentu, změna se však projeví až za zadaným příkazem, je proto nutné pokud chceme například změnit velikost prvků, zadat příkaz před prostředím které chceme změnit. Tímto způsobem se dá upravit prakticky jakákoliv vlastnost, uvedeme si tu příklady některých z nich:

Snížení tloušťky čar u bipólů - výchozí hodnota je 2

```
\ctikzset{bipoles/thickness=1}
```

Zvětšení velikosti u bipólů o 140%

```
\ctikzset{bipoles/length=1.4cm}
```

Snížení vzdálenosti proudové šipky od prvku na 20%

```
\ctikzset{current/distance = 0.2}
```

Snížení vzdálenosti šipky napětí od uzlů na 10

```
\ctikzset{voltage/distance from node=0.1}
```

Zvětšení vzdálenosti šipky napětí od prvku - výchozí nastavení je 0.08

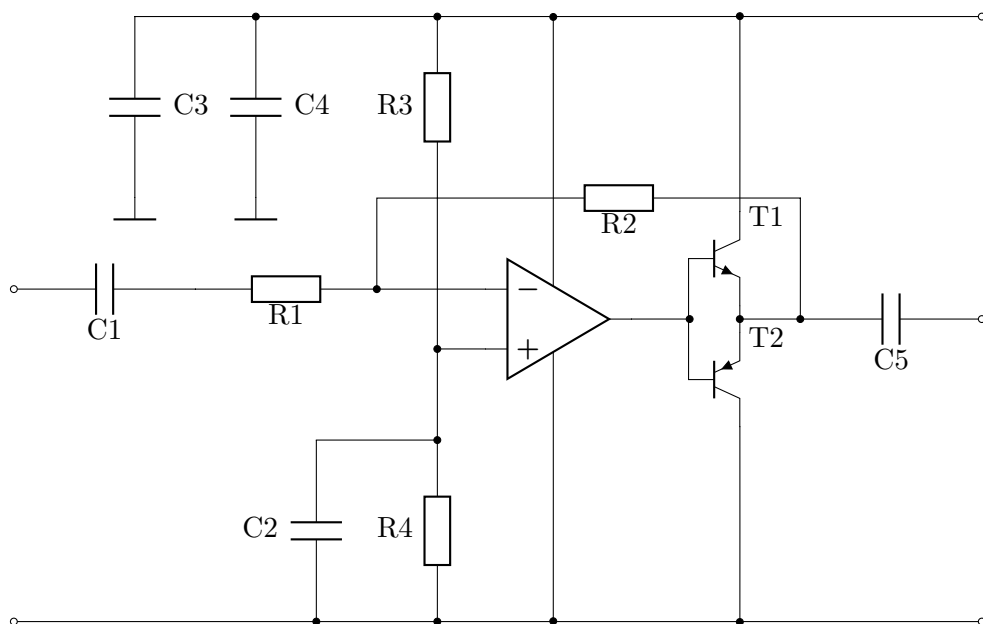
```
\ctikzset{voltage/distance from line=0.2}
```

#### 5.1.4 Sazba schéma telekomunikačních systémů

Abychom pochopili sazbu rozsáhlejších schémat s balíčkem `Circuitikz`, popíšeme si schéma nf zesilovače s operačním zesilovačem:

Vstupní signál je přiveden přes vazební kondenzátor C1 a odpor R1 na neinvertující vstup operačního zesilovače. Tento signál je v operačním zesilovači zesílen a přiveden na bázi výkonových tranzistorů T1 a T2. OZ tedy funguje jako budič koncových tranzistorů. Koncové tranzistory zesílí signál na potřebnou výstupní úroveň signálu. Celkové zesílení zesilovače jako celku je dáno odpory R1 a R2. Děličem z odporů R3 a R4 je vytvořena umělá nula pro operační zesilovač. Výstupní výkon je určen velikostí napájecího napětí a výběrem koncových tranzistorů. Zesilovač není odolný proti zkratu na výstupu.[16]

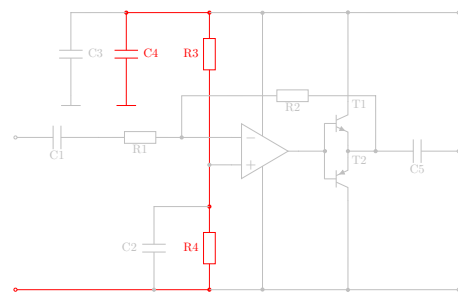
I když jsou součástí schéma více pólové prvky a vybízí se použít čistě relativní souřadnice, u rozsáhlejších schémat jako je toto, by bylo extrémně náročné a nepřehledné je použít. Použijeme proto jejich kombinaci a začneme z levého dolního rohu se souřadnicí (0,0). Snažíme se vytvářet nejvíce logické cesty z hlediska schéma i z hlediska pozdějších úprav nebo přesunů. Ohledně



Obrázek 5.4: Schéma nf zesilovače v Circuitikz

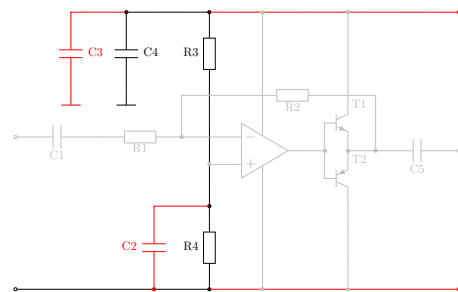
hodnot platí, že většina bipolárních prvků má velikost **3 body**, od čehož si můžeme rozvrhnout přibližné měřítko schéma.

```
\draw (0,0) to [short,o-*] (7,0)
to [R=R4,-*] (7,3)
to [short] (7,7)
to [R=R3,-*] (7,10)
to [short,-*] (4,10)
to [C=C4] ++(0,-3) node [rground]{};
```



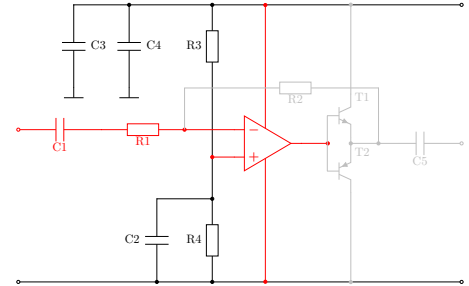
Prvním příkazem `\draw` vytvoříme kaskádu prvků: *levý kontakt - R4 - R3 a relativně pozicovaný C4 s uzemněním* - relativní souřadnice v koncové části nám ulehčí případnou pozdější manipulaci.

```
\draw (4,10) to [short] (2,10)
to [C=C3] ++(0,-3) node [rground]{};
\draw (5,0) to [C=C2,*-] (5,3) -- (7,3)
;
\draw (7,10) to [short,-o] (16,10)
(7,0) to [short,-o] (16,0);
```



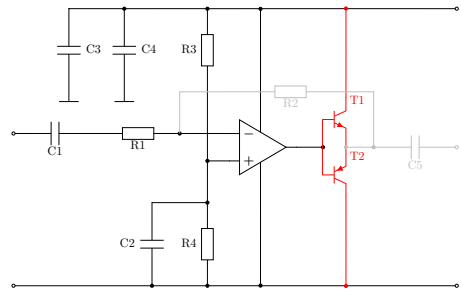
Následuje přidání paralelního C3, C2 a vodičů vedení pro lepší pozdější orientaci při sazbě.

```
\draw (9,5) node[op amp] (opamp) {}
  (opamp.+) to [short,-*] ++(-0.8,0)
  (opamp.-) to [short,-*] ++(-1.8,0)
    to [R=R1] ++(-3,0)
    to [C=C1,-o] ++(-3,0)
  (opamp.up) to [short,-*] ++(0,4.45)
  (opamp.down) to [short
    ,-*] ++(0,-4.45)
  (opamp.out) to [short,-*]
    ++(0.97,0);
```



Na absolutní pozici umístíme OZ a relativně připojíme jeho + pin, - pin ke kaskádě:  $R1 - C1$  a levý horní kontakt, napájecí up a down a výstupní pin pro připojení T1 a T2. Piny je nutné připojovat relativně, neboť neznáme jejich přesné souřadnice.

```
\draw (12,6) node[npn] (t1) {T1}
  (12,4) node[pnp] (t2) {T2}
  (t1.B) -- (t2.B)
  (t1.E) -- (t2.E)
  (t1.C) to [short,-*] ++(0,3.23)
  (t2.C) to [short,-*] ++(0,-3.23);
```



Vytvoříme NPN tranzistor T1, PNP T2, propojíme vodiči jejich báze, emitory a kolektory připojíme relativně k vedení.

```
\draw (16,5) to [C=C5,o-*] (13,5) to [
  short,-*] (12,5);
\draw (13,5) to [short] (13,7)
  to [R=R2] (7,7) -- (6,7) ----+
    (0,-1.5);
```

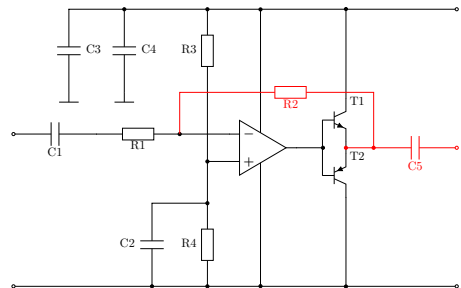


Schéma dokončíme přidáním výstupního C5 s kontaktem a R2.

## 5.2 Latex Circuit Diagram

LCD je dle mého názoru nejlepší makro v čistě absolutních souřadnicích. Nabízí velice přehlednou syntaxi doplněnou oproti makru Eltex o automatické popisky komponent a oproti balíčku

CircDia má zase kratší a jednodušší příkazy. S makrem se mi osobně pracovalo velmi dobře, jako druhý balíček si proto popíšeme právě ono.

Jak již bylo zmíněno, pro zprovoznění balíčku si jej stačí stáhnout ze stránek GoogleCode, umístit soubor do kořenového(nebo jiného) adresáře projektu a nahrát pomocí příkazu `\input`.

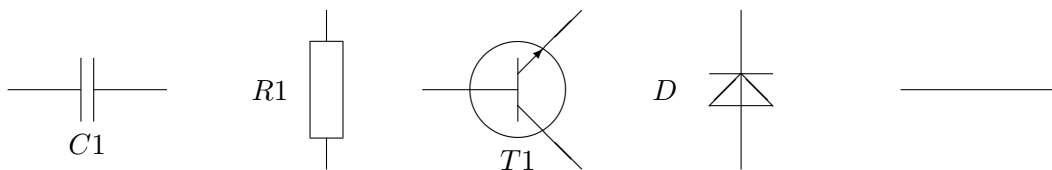
Pro práci s balíčkem je nutné mít otevřený zdrojový  $\text{\TeX}$  soubor s popisem všech dostupných komponent s jednoduchým příkladem použití balíčku a stejnojmenný PDF soubor, který je výstupem zkompilovaného  $\text{\TeX}$  souboru.

Sázíme do  $\text{\LaTeX}$  prostředí `picture`, kterému nastavíme velikost podle sázeného schéma, případně offset neboli posun celého schéma na ose  $XY$ .

```
\begin{picture}(velikostX,velikostY)(offsetX,offsetY)
...
Latex Circuit Diagram kod
...
\end{picture}
```

### 5.2.1 Sazba a spojování prvků

Vložení každého prvku se skládá z jednoduchého příkazu skládajícího se z **označení** a **absolutních souřadnic**. U elektrických součástek ještě **názvu nebo hodnoty**, popřípadě u vodičů **délkou**. U označení je zadána také orientace prvku - u horizontálních nebo vertikálních písmeny **H**, **V** například u RLC, zdrojů nebo vodičů. U komponent které vyžadují orientaci ve čtyřech směrech, jsou směry popsány písmeny **R**, **L**, **U**, **D** (Right, Left, Up a Down) například u polovodičových diod, šipek, OZ nebo značení proudů a napětí.



Každý komponent má také ve zdrojovém textu vyznačený nulový bod prvku, označený uzlem `\cn`, který značí pozici prvku na dosazených souřadnicích. U většiny prvků platí, že nulový bod je umístěn při horizontálním umístění na konci levého vodiče k prvku a při vertikálním umístění na konci dolního vodiče k prvku.

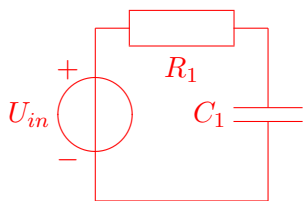
### 5.2.2 Přizpůsobení

Jelikož balíček využívá pro sazbu prostředí `picture`, můžeme jej využít pro změnu barev celého schéma nebo některé jeho části. Využijeme k tomu příkaz `\color{}`, který můžeme umístit před

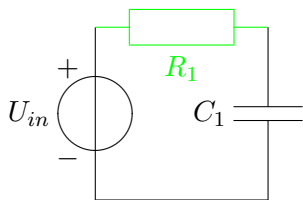
```
\setlength{\unitlength}{1.2pt} % nastaveni velikosti jednotek na 1.2pt
\begin{picture}(330,50)(0,0) % vytvoreni prostreni picture 330x50 jednotek
\hc{0,25}{C1} % horizontalni kondenzator
\vz{100,0}{R1} % vertikalni rezistor
\npnc{130,25}{T1} % npn tranzistor
\ud{230,0}{D} % dioda s orientaci nahoru
\hln{280,25}{50} % horizontalni vodice s delkou 50 jednotek
\end{picture}
```

Výpis 5.7: Příklady syntaxe a orientace prvků

první prvek pro obarvení celého schéma nebo jej vložit do závorky s jedním nebo více prvky pro obarvení pouze prvků v závorce. K tomuto přizpůsobení je nutné využít knihovnu `xcolor`.



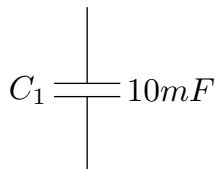
```
\begin{picture}(50,50)(0,0)
\color{red}
\vst{0,0}{U_{in}}
\vc{50,0}{C_1}
\hz{0,50}{R_1}
\hln{0,0}{50}
\end{picture}
```



```
\begin{picture}(50,50)(0,0)
\vst{0,0}{U_{in}}
\vc{50,0}{C_1}
{\color{green}\hz{0,50}{R_1}}
\hln{0,0}{50}
\end{picture}
```

V případě že budeme potřebovat pozměnit existující prvek, například změnit umístění textového pole nebo přidat nové textové pole zvlášť pro hodnotu prvku, můžeme vcelku snadno upravit popis zvolené komponenty v souboru `latex-macro.tex`. Jelikož se bude jednat o změnu prvku v knihovně, tato změna se projeví u všech těchto prvků ve schéma.

Nejprve si v souboru vyhledáme příkaz pro vytvoření požadované komponenty, složený z jednoduchých tvarů jako jsou čáry, kružnice, šipky a textové pole. Každý tvar začíná příkazem `\put(x,y)`, kde `x` a `y` jsou souřadnice daného tvaru relativně k nulovému bodu prvku. Editací těchto souřadnic tedy můžeme s jednotlivými tvary pohybovat. Jako příklad si uvedeme již



```
\newcommand{\vc}[3] % Uprava na 3 vstupni parametry
{\put(#1){\begin{picture}(32,50)
\put(0,0){\line(0,1){23}}
\put(0,27){\line(0,1){23}}
\put(-10,23){\line(1,0){20}}
\put(-10,27){\line(1,0){20}}
\put(-12,25){\makebox(0,0)[r]{$\{#2\}$}} % Popisek #2
\put(12,25){\makebox(0,0)[l]{$\{#3\}$}} % Pridani popisku #3
\end{picture}}}
```

Obrázek 5.5: Úprava vertikálního kondenzátoru pro druhý popisek

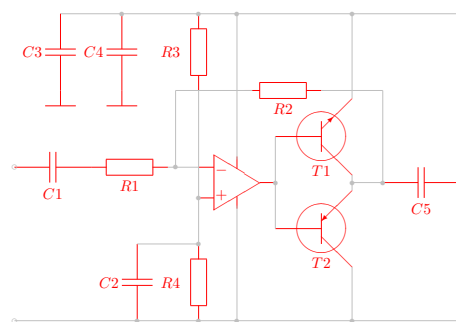
zmíněné přidání druhého popisku pro hodnotu vertikálního kondenzátoru C15.5.

### 5.2.3 Sazba schéma telekomunikačních systémů

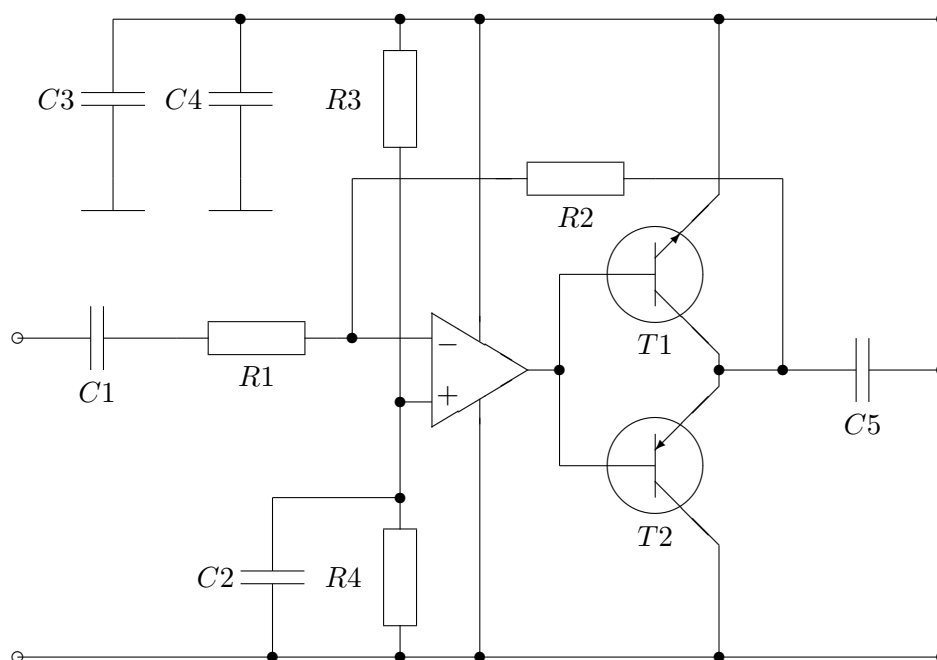
Sazba rozsáhlejších schémat mi přišla u LCD podstatně jednodušší a rychlejší než u balíčku Circuitkz. Osvědčilo se mi totiž první vytvořit jednotlivé komponenty, přesně je umístit do pracovní plochy a teprve potom je propojit vodiči a nakonec uzly nebo kontakty.

Jako první si rozvrhneme jak velkou budeme asi potřebovat pracovní plochu a pak do ní začneme sázet jednotlivé komponenty. Pro snadnější určení měřítka schéma - většina prvků je dlouhá 50 jednotek, tzn. že když vytvoříme například horizontálně orientovaný rezistor na pozici (0,0), další prvek bude začínat na pozici (50,0).

<code>\vz{120,0}{R4}</code>	<code>\vo{120,80}{}</code>
<code>\vc{80,0}{C2}</code>	<code>\npnc{170,120}{T1}</code>
<code>\vz{120,150}{R3}</code>	<code>\pnpc{170,60}{T2}</code>
<code>\hc{0,100}{C1}</code>	<code>\hc{240,90}{C5}</code>
<code>\hz{50,100}{R1}</code>	<code>\hz{150,150}{R2}</code>
<code>\vc{30,150}{C3}</code>	<code>\hgp{70,150}</code>
<code>\vc{70,150}{C4}</code>	<code>\hgp{30,150}</code>

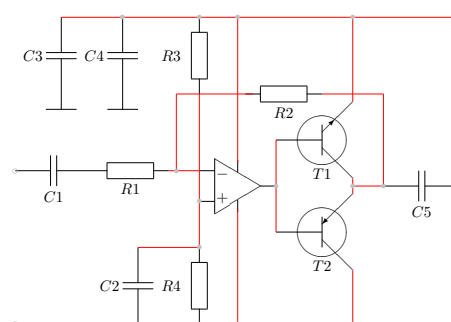


Jak již bylo avizováno, začneme hrubým rozhozením jednotlivých komponent na jejich přibližné pozice a poté je můžeme přesněji pozicovat. U daného schéma se mi také osvědčilo použít vodiče vedení jako šablonu pro lepší představu výsledného schéma.



Obrázek 5.6: Schéma nf zesilovače v Latex Circuit Diagram

<code>\vln{120,50}{100}</code>	<code>\vln{145,105}{95}</code>
<code>\vln{170,60}{60}</code>	<code>\hln{0,0}{290}</code>
<code>\vln{220,0}{35}</code>	<code>\hln{30,200}{260}</code>
<code>\vln{220,145}{55}</code>	<code>\hln{80,50}{40}</code>
<code>\vln{220,85}{10}</code>	<code>\hln{100,100}{20}</code>
<code>\vln{105,100}{50}</code>	<code>\hln{220,90}{20}</code>
<code>\vln{240,90}{60}</code>	<code>\hln{105,150}{50}</code>
<code>\vln{145,0}{75}</code>	<code>\hln{200,150}{40}</code>



Následuje přidání vodičů které si můžeme rozdělit podle přehlednosti na horizontální a vertikální, zleva doprava, abychom se i pro pozdější úpravy ve schéma vyznali. Přesné souřadnice pro dosazení například na piny OZ nebo tranzistorů můžeme najít ve zdrojovém souboru knihovny u dané komponenty.

<code>\out{0,0}</code>	<code>\cn{220,0}</code>
<code>\out{290,0}</code>	<code>\cn{70,200}</code>
<code>\out{0,100}</code>	<code>\cn{120,200}</code>
<code>\out{290,90}</code>	<code>\cn{220,200}</code>
<code>\out{290,200}</code>	<code>\cn{145,0}</code>
<code>\cn{80,0}</code>	<code>\cn{145,200}</code>
<code>\cn{120,0}</code>	<code>\cn{170,90}</code>
<code>\cn{120,50}</code>	<code>\cn{220,90}</code>
<code>\cn{120,80}</code>	<code>\cn{240,90}</code>
<code>\cn{105,100}</code>	

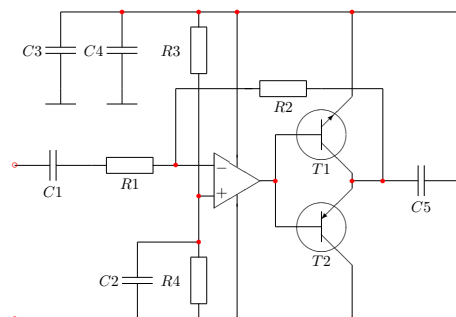


Schéma dokončíme přidáním uzlů a kontaktů. Opět se vyplatí do budoucna udělat si logickou posloupnost příkazů. Jelikož LCD používá poměrně velké jednotky sazby, je u větším schémata nutné pomocí příkazu `\setlength{\unitlength}{}` upravit velikost jednotek pro A4 sazbu na 1-1,5pt.



## 6 Závěr

Cílem bakalářské práce bylo popsat možnosti sazby elektronických schémat s balíčkem PGF/-TikZ. Bohužel balíčky běžící pod touto knihovnou jsou pouze dva a to Circuitikz a Circuits, který z něj vychází a obě knihovny se liší pouze nepatrně. Musel jsem proto rozšířit výběr balíčků i na ty které vychází z prostředí `picture` nebo PostScriptu. Totéž platí i pro popsané WYSIWYG editory, kdy čistě pod PGF/TikZ běží pouze TikzEdt s knihovnou Circuits.

Před začátkem práce jsem neměl se sazbou elektronických schémat žádnou zkušenost a mohu tedy práci s jednotlivými balíčky hodnotit očima laika jako doporučení pro začínající uživatele.

### 6.1 Circuitikz

Jedná se nejspíše o nejlepší balíček co se týče možností sazby, přizpůsobení jednotlivých komponent a dostupné kvalitní dokumentace s popsanými všemi dostupnými prvky a příklady používání. Velkou výhodou je i jeho rozšířenost mezi uživateli na internetu, kde nalezneme spoustu rad, postupů nebo i instruktážních videí.

Knihovna nabízí sazbu jak v absolutních, tak relativních souřadnicích. Její syntaxe je poměrně jednoduchá a po několika schématech intuitivní a vhodná pro začínající uživatele.

### 6.2 Circuits

Knihovna Circuits je komponentou balíčku TikZ, která vychází z předchozí knihovny Circuitikz. Jedná se knihovnu pracující jak v absolutních, tak i relativních souřadnicích, syntaxí velice podobnou Circuitikz.

Největší nevýhodou knihovny je velice slabá dokumentace na několika stránkách manuálu k PGF/TikZ, která je dle mého názoru pro začínající uživatele naprosto nedostatečná a tím pádem bych raději doporučil Circuitikz.

### 6.3 elTeX

Eltex je příjemné a velice jednoduché makro pracující pouze v absolutních souřadnicích. Nabízí dokumentaci se všemi dostupnými prvky a příklady použití, kde bych jen trochu vytknul absenci náhledových obrázků k prvkům.

Makro je díky své jednoduchosti velmi vhodné pro začátečníky. Nevýhodou u absolutních knihoven obecně je nutnost časté kompilace a kontrolování souřadnic výsledného schéma nejlépe s pomocnou mřížkou.

## 6.4 Circ

Circ je balíček, který mě zaskočil svou poměrně složitou instalací. Pracuje v čistě relativních souřadnicích a jeho přehledná dokumentace obsahuje popsané všechny dostupné prvky a syntaxi, která je dle mého názoru složitější a popsaná pouze na jednom příkladu.

Měl jsem proto a pořád mám s balíčkem při práci určité nejasnosti a začínajícím uživatelům bych jej spíše nedoporučoval. Nicméně uživatelé kterým bude vyhovovat, ocení jeho krátké a jednoduché příkazy a rychlé pozdější úpravy schéma.

## 6.5 pst-circ

Pst-circ je balíček který nás přivítá rychlou instalací a velmi pěknou a přehlednou dokumentací, která obsahuje přehled všech komponent, jejich syntaxi a spoustu názorných příkladů. Pracuje v absolutních pozicích jako např. elTeX, ale narozdíl od něj mi přišla jeho sytaxe elegantnější a připravená pro pozdější rychlé zásahy nebo změny souřadnic ve schématu. Jedná se o balíček, který je díky velkému množství příkladů snadno pochopitelný a tudíž vhodný i pro začínající uživatele.

Pro mě byla největší nevýhoda tohoto balíčku jelikož pracuji v MikTeXu, nutnost kompilace v kompilátoru XeLaTeX, kde trvá kompilace poměrně déle a je nutné kompilovat v externím souboru. Pro instalace TeX Live však není problém. Další menší nevýhodou je dle mého názoru lehce amatérský vzhled některých sázených komponent.

## 6.6 Latex Circuit Diagram

Jedná se o makro syntaxi velmi podobné balíčku elTeX. Dokumentace obsahuje TeX soubor s přehledem všech prvků a příkladem použití a PDF soubor s jeho kompilací pro rychlý náhled jak který kus kódu vypadá. Makro je stejně jako elTeX na používání velmi jednoduché a rychle pochopitelné. Přišlo mi ze všech čistě absolutních balíčků nejlepší, právě proto jsem ho podrobně popsal.

## 6.7 CircDia

CircDia je poměrně mladý balíček syntaxi opět podobný balíčku elTeX nebo LCD. Nabízí PDF dokumentaci prozatím pouze v německém jazyce, nicméně s velmi přehledně popsanými všemi prvky a jejich různými variantami sazby s jedním příkladem použití, který však pro plné pochopení sytaxe plně dostačuje. Stejně jako předchozí balíčky s možností sazby v absolutních souřadnicích je vhodný i pro úplné začátečníky.

Možností sazby elektrických schémat v systému L<sup>A</sup>T<sub>E</sub>X a PGF/TikZ jsou velké. A ať už budeme chtít sázet s jedním z popsaných balíčků nebo sáhneme po jednodušším WYSIWYG

editoru, věřím že si každý najde svůj způsob, který mu bude vyhovovat jak z hlediska rychlosti sazby, tak z hlediska celkového vzhledu schéma.

## 7 Reference

### Literatura

- [1] Wikipedia. *'LaTeX'*, *Wikipedia, The Free Encyclopedia*, 2018.  
<https://en.wikipedia.org/w/index.php?title=LaTeX&oldid=845817585>
- [2] Schenk, Christian. *About MiKTeX - MiKTeXorg. Home - MiKTeXorg*, 2018.  
<https://miktex.org/about>
- [3] Redaelli, A., Massimo. *Circuitikz 0.8.3 manual*, 2017.  
<http://texdoc.net/texmf-dist/doc/latex/circuitikz/circuitikzmanual.pdf>
- [4] Tantau, Till. *The TikZ and PGF Packages*, 2015.  
<http://ftp.cvut.cz/tex-archive/graphics/pgf/base/doc/pgfmanual.pdf>
- [5] Gajdosik, Libor. *CTAN: Package eltex*, 2009.  
<https://ctan.org/pkg/eltex>
- [6] Tannert, Sebastian. *Package CIRC*, 2006.  
<http://ftp.cvut.cz/tex-archive/macros/generic/diagrams/circ/circ.pdf>
- [7] Voß, Herbert. *pst-circ – PSTricks package for drawing electric circuits*, 2017.  
<http://ftp.cvut.cz/tex-archive/graphics/pstricks/contrib/pst-circ/doc/pst-circ-doc.pdf>
- [8] Wikipedia. *PostScript* - *Wikipedia*, 2009.  
<https://en.wikipedia.org/wiki/PostScript>
- [9] Jak psát web. *Délkové jednotky v CSS*, 2017.  
<https://www.jakpsatweb.cz/css/css-jednotky.html>
- [10] Silvonon, Kimmo. *Google Code Archive*, 2011.  
<https://code.google.com/archive/p/latex-circuit-diagram/>
- [11] Krause, Stefan. *CircDia – a LaTeX package for drawing circuit diagrams*, 2018.  
<http://www.taylorgruppe.de/circdia/>
- [12] Wikibooks. *LaTeX/Picture* - *Wikibooks, open books for an open world*, 2018.  
<https://en.wikibooks.org/wiki/LaTeX/Picture>
- [13] Wikipedia. *WYSIWYG* - *Wikipedia*, 2010.  
<https://en.wikipedia.org/wiki/WYSIWYG>

## Reference

---

- [14] Wikipedia. *Inkscape - Wikipedia*, 2010.  
<https://en.wikipedia.org/wiki/Inkscape>
- [15] Stack Exchange. *TeX - LaTeX Stack Exchange*, 2018.  
<https://tex.stackexchange.com/>
- [16] Oklike.c-a-v.com. *KV RX*, 2018.  
<http://oklike.c-a-v.com/soubory/1rx.htm>

## 8 Přílohy

Obsah přiloženého CD

Figures.....	Složka obsahující externí obrázky k práci
Files .....	Složka s externími, zkompilevanými schématy v PDF
Schema .....	Spustitelné schéma v TeX souborech použité v práci
WYSIWYG.....	Zdrojové soubory WYSIWYG editorů s jejich exportem
diploma-440.cls.....	Soubor nezbytný pro správnou kompilaci práce
latex-macro.tex .....	Knihovna balíčku LCD, nutná pro kompilaci práce
main.pdf .....	Výstupní soubor zkompilevaného main.tex
main.tex.....	Zdrojový soubor práce